

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BOARD OF PATENT APPEALS AND INTERFERENCES**

Applicant: )  
BORIS MAYER ET AL. )  
 )  
Serial No.: 10/524,063 )  
 )  
Title: METHOD AND SYSTEM )  
FOR TRANSMITTING )  
NOTIFICATIONS TO USERS )  
OF A LOGISTIC SYSTEM )  
 )  
Filed: AUGUST 6, 2003 )  
 )  
Group Art Unit: 3687 )  
 )  
Examiner: STEVEN J. MALONE )  
 )  
Confirmation No. 4442 )

**APPEAL BRIEF**

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

This brief contains items under the following headings as required by 37

C.F.R. § 41.37 and M.P.E.P. § 1206:

- |       |   |
|-------|---|
| I.    | Real Party In Interest                        |
| II.   | Related Appeals and Interferences             |
| III.  | Status of Claims                              |
| IV.   | Status of Amendments                          |
| V.    | Summary of Claimed Subject Matter             |
| VI.   | Grounds of Rejection to be Reviewed on Appeal |
| VII.  | Argument                                      |
| VIII. | Claims Appendix                               |
| IX.   | Evidence Appendix                             |
| X.    | Related Proceedings Appendix                  |

Appendix A Claims

Appendix B References Cited by the Examiner

Tilles et al., U.S. Patent No. 6,748,295 ("Tilles")

Gustafsson, U.S. Patent No. 6,424,841 ("Gustafsson")

**I. Real Party In Interest**

The real party in interest is Deutsche Post AG, the assignee of the entire right, title, and interest to this application as evidenced by the assignment document recorded at reel 016352, frame 0580.

**II. Related Appeals and Interferences**

There are no related appeals or interferences.

**III. Status of the Claims**

**A. Total Number of Claims in the Application**

There are five claims in the application.

**B. Current Status of the Claims**

1. Claims canceled: 1-17
2. Claims pending: 18-22
3. Claims allowed: None
4. Claims withdrawn: None
5. Claims rejected: 18-22

**C. Claims on appeal**

The claims on appeal are claims 18-22.



**IV. Status of the Amendments**

There are no outstanding amendments to the application.

**V. Summary of the Claimed Subject Matter**

Although specification citations are inserted below in accordance with 37 C.F.R. § 41.37, these reference numerals and citations are merely examples of where support may be found in the specification for the terms used in this section of the brief. There is no intention to in any way suggest that the terms of the claims are limited to the examples in the specification. Although, as demonstrated by the reference numerals and citations below, the claims are fully supported by the specification as required by law, it is improper under the law to read limitations from the specification into the claims. In short, the reference numerals and specification citations are not to be construed as claim limitations or in any way used to limit the scope of the claims.

The invention, as defined in claims 18-21, and with reference to FIGS. 1-4, is a method for transmitting notifications to users of a logistic system, said logistic system comprising at least one parcel compartment system with at least one registered user, wherein notification orders are transmitted to a central sending component 30 which, based on the notification orders, accesses at least one database 70, 80, 100, and generates and sends appropriate notifications to the user. See the instant application, page 1, lines 7-8; page 3, line 4; page 5, lines 1-6; and page 5, lines 8-13. The method comprises calling up different modules with associated functions in response to different events within the logistic system, said modules being selected from the group consisting of a client database 70, a registration unit, and a system administration unit. *Id.* at page 5, lines 1-6. The method further comprises, generating notification orders by the modules, writing the notification orders into a communication request queue 40 so the notification orders can be sent in a deferred manner, and reading the notification orders from the

communication request queue 40 by a queue reader 50 in a timer-controlled manner and transferring the notification orders to the central sending component 30. *Id* at page 5, line 21 through page 7, line 27. The method further comprises, generating appropriate user-specific notifications by the central sending component, and sending said notifications to the user by the central sending unit<sup>1</sup> via a gateway 120, wherein said generating comprises accessing at least one client database 70, a parcel database 80, an automatic parcel delivery machine database 90, and a document database 100 by the central sending component 30, wherein said method further includes validating the status of the notification orders in a delivery contract logic 20 before transferring the notification orders to the central sending component 30. *Id* at page 10, lines 12-21; and page 23, lines 5-12.

The invention as defined in claim 22 is a device for transmitting notifications to users of a logistic system that operates one or more parcel compartment systems, wherein the logistic system comprises modules having functions for generating notification orders, of a central sending component 30, of a communication request queue 40 for storing the notification orders so that the notification orders can be sent in a deferred manner, of a document database 100 with templates 110 for generating individual notifications for specific users, of a client database 70 with information about clients, of a parcel database 80 with information about parcels, of an automatic parcel delivery machine database 90 with information about parcel compartment systems, and of a gateway 120 for sending notifications, wherein the modules are one or more of a client database 70, a registration unit and a system administration

---

<sup>1</sup> Typographical error in claim 18, should read central sending “component.” Applicants respectfully request correction via an examiner’s amendment if the claimed subject matter is found to be allowable.

unit for the logistic system. *Id* at page 5, lines 1-13; page 5, line 21-page 7, line 27;  
page 10, lines 4-10 and 12-21; and page 23, lines 5-12.

**VI. Grounds of rejection to be reviewed on appeal.**

The grounds of rejection to be reviewed on appeal are:

- 1) Are each of claims 18-21 allowable over Tilles in view of Gustafsson?
- 2) Is claim 22 allowable over Tilles in view of Gustafsson?

VII. Argument

**A. Claims 18-22 meet the requirements for patentability and are allowable over Tilles in view of Gustafsson.**

Claims 18-21

Claim 18 is an independent claim and claims 19-21 depend directly therefrom.

Independent claim 18 recites a method for transmitting notifications to users of a logistic system comprising, in part, writing notification orders into a communication request queue so the notification orders can be sent in a deferred manner and reading the notification orders from the communication request queue by a queue reader in a timer-controlled manner and transferring the notification orders to the central sending component.

The Patent Office “has the burden under § 103 to establish a prima facie case of obviousness.” *In re Fine*, 837 F.2d 1071, 1074 (Fed. Cir. 1988); MPEP § 2142 (8<sup>th</sup> Ed., Rev. 6, Sept. 2007) (“The examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness.”). The Supreme Court recently identified a number of rationales that may be used to support a conclusion of obviousness, consistent with the framework set forth in its decision in *Graham v. John Deere. Co.* See *KSR Int’l Co. v. Teleflex Inc.*, 127 S.Ct. 1727, 1739-40 (2007). These and other representative rationales are described at MPEP § 2143 (8<sup>th</sup> Ed., Rev. 6, Sept. 2007). Regardless of the supporting rationale, however, the Patent Office must clearly articulate facts and reasons why the claimed invention “as a whole” would have been obvious to a hypothetical person having ordinary skill in the art at least as of the claimed invention’s effective filing date. See *KSR Int’l*, 127 S.Ct. at 1741 (citing with approval *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (“[R]ejections on obviousness grounds cannot be sustained by mere conclusory

statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.”)); see *also* MPEP § 2143 (“The key to supporting any rejection under 35 USC § 103 is the clear articulation of reason(s) why the claimed invention would have been obvious.”).

The sole issue on appeal is whether the final action established a *prima facie* case of obviousness. Applicants respectfully submit that the final action failed to establish a *prima facie* case of obviousness for at least three reasons: 1) the final action fails to clearly articulate reasons why the claimed invention would have been obvious because the final action is incomplete and confusing, 2) the final action fails to show where the prior art discloses each and every claim limitation, and 3) the alleged motivation for combining the teachings of the prior art is not reasonable.

Applicants respectfully submit that the final action is incomplete and confusing and thus does not establish a *prima facie* case of obviousness. In particular, the final action alleges on page 3 that Tilles discloses “writing the notification orders into a communication request queue . . . so the orders can be sent in a deferred manner.” Then, on page 5, the final action concedes that Tilles “fails to explicitly disclose writing the notification orders into a communication request queue so the orders can be sent in a deferred manner.” In other words, the final action alleges Tilles discloses a particular claim limitation and later concedes that Tilles fails to disclose the same claim limitation. Applicants are confused as to what exactly is alleged by the final action that Tilles discloses in this regard. For this reason alone, the final rejection fails to establish a *prima facie* case of obviousness and the rejection of claims 18-21 should be withdrawn.

Regardless of the allegation in the final action on page 3 that Tilles discloses “writing the notification orders into a communication request queue . . . so the orders

can be sent in a deferred manner,” Applicants respectfully submit that this allegation is completely without support. The final action alleges at page 3 that the limitation of “writing the notification orders into a communication request queue” is supported by a disclosure in Tilles of “a software instruction queue, (see Abstract, via application software; or a database used for queuing email notifications).” Applicants respectfully submit that the “application software” disclosed in the abstract of Tilles has nothing to do with queuing email notifications. Rather, the “application software” disclosed in the abstract of Tilles relates to “implementing an application interface of selectively configurable Active X controls for **providing user access**, such as an employee of a delivery service company and/or a customer of the delivery service company and customer access **to one or more storage bins** located behind a set of normally closed doors, for providing access control to the bins, and for managing the location of the items in the storage subsystem.” Emphasis added, see Tilles Abstract. Nowhere is email even mentioned in this section, let alone queuing email notifications. For this additional reason, the final action fails to establish a prima facie case of obviousness and Applicants respectfully request the rejection of claims 18-21 be withdrawn.

Likewise, at page 3 the final action alleges that Tilles discloses “reading the orders from the communication request queue (program instruction queue) by a queue reader (memory device) in a timer-controlled manner (a scanner which includes a timer based central processing unit CPU or microprocessor).” Applicants respectfully submit that this allegation is completely without support. Applicants have studied Tilles and were unable to find the alleged “program instruction queue,” “memory device,” or “a scanner which includes a timer based central processing unit CPU or microprocessor.” In fact, Applicants were unable to locate the words



“program instruction queue,” “memory device,” “timer based,” “central processing unit,” or “microprocessor” in the specification or drawings of Tilles. For this additional reason, the final action fails to establish a prima facie case of obviousness and Applicants respectfully request withdrawal of the rejection of claims 18-21.

Applicants respectfully submit that the alleged motivation to modify Tilles is not reasonable. The final action alleges at page 5 that “Gustafsson discloses a short message service with improved utilization of available bandwidth” and thus, “it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the item delivery and retrieval system of Tilles et al. to include writing notification orders into a communication request queue so the order can be sent in a deferred manner as taught by Gustafsson.” However, Gustafsson teaches his technique is necessary because “[o]ne-way-SMS represents a narrowband channel that can carry data in primarily one direction, with acknowledgements going in the opposite direction.” See Gustafsson col. 2, lines 8-10.

The problem solved by Gustafsson is to “more efficiently utilize the available transmission bandwidth in a [narrowband channel] wireless network.” *Id* at col. 2, lines 56-57. Tilles does not teach or suggest utilizing a narrowband channel for transmitting e-mail messages. Rather, Tilles implies the opposite, using a broadband channel. Tilles teaches the option of users “requesting internet e-mail notification service.” See Tilles col. 13, lines 38-39. One of ordinary skill in the art would understand that the “internet e-mail notification service” is a broadband service because Tilles teaches that his system is connected to the internet and that the application configurable software installed on the Tilles system is “internet web page based.” See Tilles col. 3, lines 54-55.

One of ordinary skill in the art would choose a broadband channel to transmit and receive internet web pages due to the amount of information contained in web pages. Nowhere does Tilles even contemplate using a narrowband channel of any sort to transmit e-mail. Thus, the final action alleges a motivation to combine to solve a problem that does not exist in Tilles, and one of ordinary skill in the art would not perceive a need to optimize the Tilles system because Tilles teaches using a broadband channel. As a result, the alleged motivation to modify Tilles with the teachings of Gustafsson (i.e., to improve utilization of available bandwidth) is not reasonable. For this additional reason, the final action fails to establish a prima facie case of obviousness and Applicants respectfully request withdrawal of the rejection of claims 18-21.

#### Claim 22

Claim 22 is an independent claim. Independent claim 22 recites a device for transmitting notifications to users of a logistic system comprising, in part, a communication request queue for storing the notification orders so that the notification orders can be sent in a deferred manner. The arguments above with respect to claims 18-21 apply to claim 22 as well.

Although the rejection of claim 22 is found under the heading of "Claims 18-22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Tilles et al. (6,748,295) in view of Gustafsson (6,424,841)," the text of the official action beginning at page 6, setting forth particular reasons for the rejection of claim 22, fails to identify any deficiency in Tilles that Gustafsson cures (i.e., Tilles is the only reference cited in the section rejecting claim 22 specifically). Nor does the official action allege that claim 22 is obvious over Tilles alone. Thus, the final action fails to establish a prima facie case of obviousness because it appears the section rejecting

claim 22 is incomplete. For this reason alone, Applicants respectfully request withdrawal of the rejection of claim 22.

The final action also fails to establish a prima facie case of obviousness of claim 22 because the final action fails to show that Tilles discloses each and every claim limitation. The final action fails to even allege that Tilles discloses or suggests “a communication request queue **for storing the notification orders so that the notification orders can be sent in a deferred manner,**” (emphasis added) as recited by claim 22. Rather, the final action only alleges that Tilles discloses “a communication request queue.” See the final action at page 7. Simply alleging “a communication request queue” is not sufficient to show that Tilles teaches “a communication request queue for storing the notification orders so that the notification orders can be sent in a deferred manner.” Moreover, the final action concedes at page 5 that “Tilles et al. fails to explicitly disclose writing the notification orders into a communication request queue so that the orders can be sent in a deferred manner.” For this additional reason, the final action fails to establish a prima facie case of obviousness and Applicants respectfully request withdrawal of the rejection of claim 22.

**VIII. Claims Appendix**

An appendix containing a copy of the claims involved in the appeal is attached as Appendix A hereto.

**IX. Evidence Appendix**

Copies of each reference relied upon by the examiner in his rejections are submitted at Appendix B.

Serial No. 10/524,063

Atty. Docket No. 30882/DP019

Appeal Brief dated March 11, 2009

Appeal from the Final Action Dated August 15, 2008

**X. Related Proceedings Appendix**

None.


**CONCLUSION**

In view of the foregoing remarks, it is respectfully submitted that each of claims 18-22 is allowable over the cited references. The applicants request the Board to reverse the examiner with respect to each of the rejections of the claims, and return the application to the examiner for further prosecution consistent with its decision. In the event any additional fees are necessary in connection with the present appeal, kindly charge the cost thereof to Deposit Account No. 13-2855 under Order No. 30882/DP019.

Respectfully submitted,

**March 11, 2009**

By:

  
Michael A. Chinlund, Reg. No. 55,064  
Agent for Applicants & Assignee  
Marshall, Gerstein & Borun LLP  
233 South Wacker Drive  
6300 Sears Tower  
Chicago, Illinois 60606-6357  
Tel.: (312) 474-6300  
Fax: (312) 474-0448

**APPENDIX A**

**CLAIMS APPENDIX**

The following listing of claims replaces all prior presentations or listing of claims.

18. (Previously Presented) A method of transmitting notifications to users of a logistic system,

said logistic system comprising at least one parcel compartment system with at least one registered user, wherein notification orders are transmitted to a central sending component which, based on the notification orders, accesses at least one database and generates and sends appropriate notifications to the user, the method comprising :

(a) calling up different modules with associated functions in response to different events within the logistic system, said modules being selected from the group consisting of a client database, a registration unit, and a system administration unit for the logistic system;

(b) generating notification orders by the modules;

(c) writing the notification orders into a communication request queue so the notification orders can be sent in a deferred manner; and

reading the notification orders from the communication request queue by a queue reader in a timer-controlled manner and transferring the notification orders to the central sending component;

(d) generating appropriate user-specific notifications by the central sending component; and,

(e) sending said notifications to the user by the central sending unit via a gateway;

wherein said generating comprises accessing at least one client database, a parcel database, an automatic parcel delivery machine database, and a document database by the central sending component, wherein said method further includes validating the status of the notification orders in a delivery contract logic before transferring the notification orders to the central sending component.



19. (Previously Presented) The method of Claim 18, comprising allocating client data, parcel data, and parcel compartment system data in the databases by means of identities (IDs).

20. (Previously Presented) The method Claim 18, wherein the events in the logistic system comprise at least the following:

- registration of the new user
- change in the user data
- placement of a new parcel in a parcel compartment system
- picking up a parcel from a parcel compartment system
- sending back a parcel
- adding a substitute for pick-up of a parcel
- removing a substitute.

21. (Previously Presented) The method of Claim 18, comprising sending the notifications to the users in the form of at least one of e-mail and Short Message Service (SMS).

22. (Previously Presented) A device for transmitting notifications to users of a logistic system that operates one or more parcel compartment systems, wherein the logistic system comprises modules having functions for generating notification orders, of a central sending component, of a communication request queue for storing the notification orders so that the notification orders can be sent in a deferred manner, of a document database with templates for generating individual notifications for specific users, of a client database with information about clients, of a parcel database with information about parcels, of an automatic parcel delivery machine database with information about parcel compartment systems and of a gateway for sending the notifications, wherein the modules are one or more of a

client database, a registration unit and a system administration unit for the logistic system.

Serial No. 10/524,063

Atty. Docket No. 30882/DP019

Appeal Brief dated March 11, 2009

Appeal from the Final Action Dated August 15, 2008

**APPENDIX B**

**References Cited Appendix**



US006748295B2

(12) **United States Patent**  
**Tilles et al.**

(10) **Patent No.:** **US 6,748,295 B2**  
 (45) **Date of Patent:** **Jun. 8, 2004**

(54) **ITEM DELIVERY AND RETRIEVAL SYSTEM**

5,113,351 A 5/1992 Bostic

(75) **Inventors:** **David J. Tilles**, Woodstock, MD (US);  
**David J. Janos**, Ellicott City, MD (US);  
**Mark T. Neebe**, Catonsville, MD (US);  
**Bruce G. Chestnutt**, Sykesville, MD (US);  
**Ann C. Schofield**, Ellicott City, MD (US);  
**Randall K. Neilson**, Crownsville, MD (US)

(List continued on next page.)

#### FOREIGN PATENT DOCUMENTS

EP	0 821 518 A	1/1998
EP	0 845 747 A2	1/1998
FR	2 621 803 A	4/1989
FR	2 643 479 A	8/1990

#### OTHER PUBLICATIONS

REMSTAR catalog, showing various retrieval units.  
 Hanel's carousel publication, showing automatic compartment doors on a storage/retrieval unit.

*Primary Examiner*—Donald P Walsh  
*Assistant Examiner*—Michael E. Butler

(74) *Attorney, Agent, or Firm*—Birch, Stewart, Kolasch & Birch, I.L.P.

(73) **Assignee:** **Northrop Grumman Corporation**, Los Angeles, CA (US)

(\*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) **Appl. No.:** **09/817,375**

(22) **Filed:** **Mar. 27, 2001**

(65) **Prior Publication Data**

US 2002/0032501 A1 Mar. 14, 2002

#### Related U.S. Application Data

(60) Provisional application No. 60/265,875, filed on Feb. 5, 2001, and provisional application No. 60/220,842, filed on Jul. 26, 2000.

(51) **Int. Cl.**<sup>7</sup> ..... **G06F 17/00**; **G06F 7/00**

(52) **U.S. Cl.** ..... **700/241**; **700/242**; **700/230**;  
**700/216**; **700/215**; **340/568.1**

(58) **Field of Search** ..... **700/214**; **340/569**,  
**340/825**

(56) **References Cited**

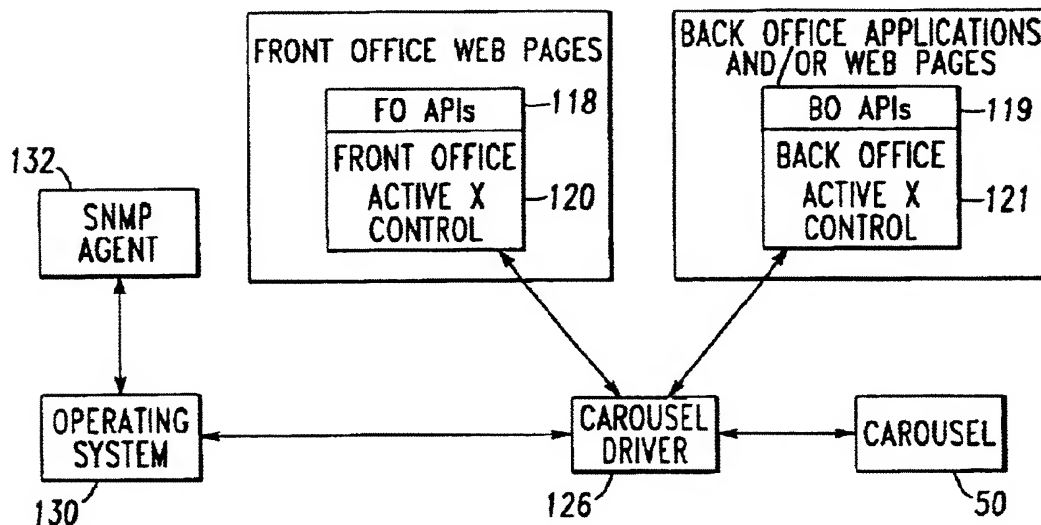
#### U.S. PATENT DOCUMENTS

4,072,825 A	*	2/1978	McLay et al.	179/18 B
4,950,118 A	*	8/1990	Mueller et al.	414/274

(57) **ABSTRACT**

An item delivery and retrieval system including a storage subsystem and a computer subsystem. The storage subsystem includes a secure enclosure having an item storage carousel including internal controller apparatus. The computer subsystem is embodied in internet web page based customized application software for implementing an application interface of selectively configurable ActiveX controls for providing user access, such as an employee of a delivery service company and/or a customer of the delivery service company and customer access to one or more storage bins located behind a set of normally closed doors, for providing access control to the bins, and for managing the location of the items in the storage subsystem. The doors are opened when proper identification is provided by the customer so as to permit retrieval of items located in specifically designated bin(s) or to return items thereto.

**35 Claims, 8 Drawing Sheets**



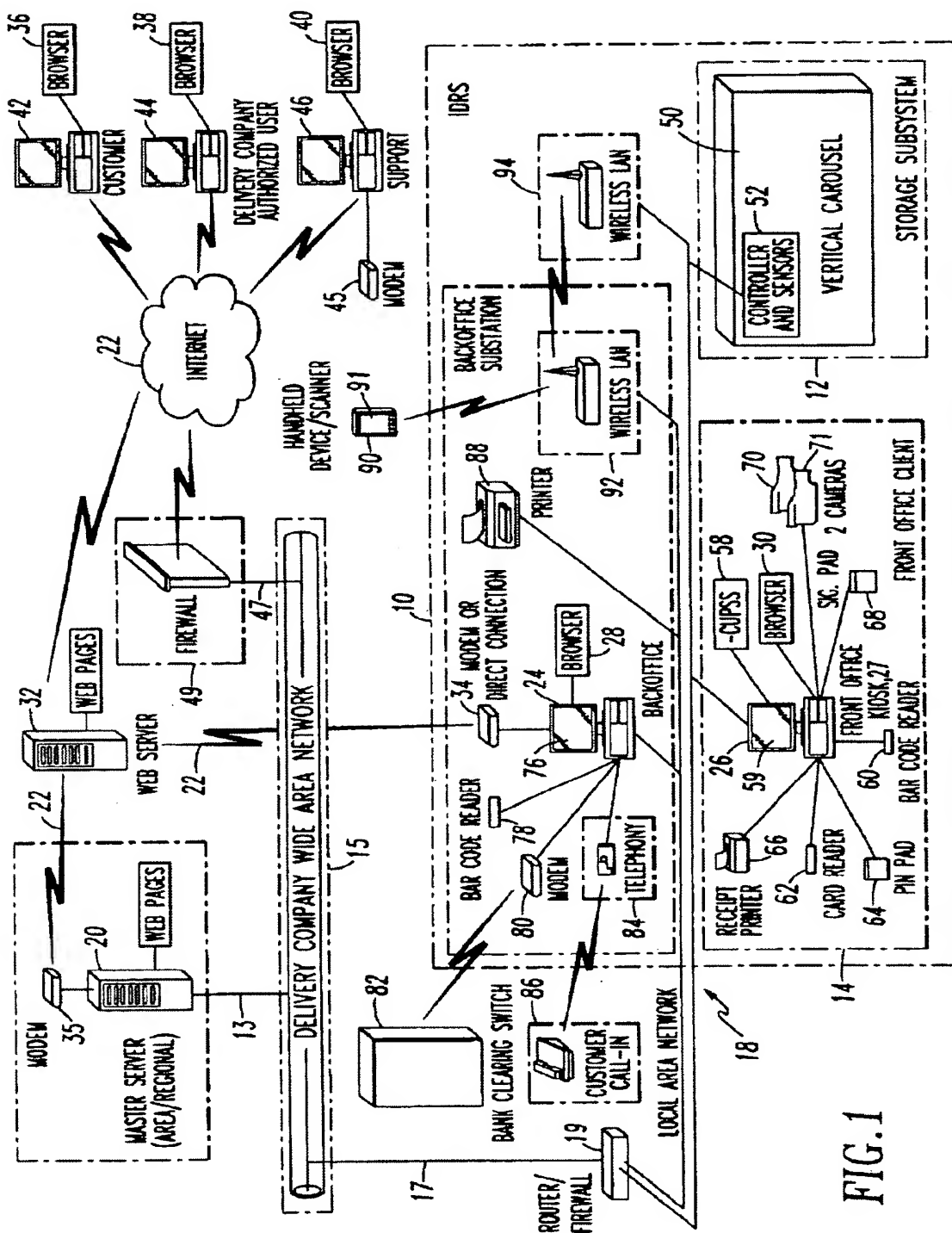
# US 6,748,295 B2

Page 2

## U.S. PATENT DOCUMENTS

5,273,183	A	*	12/1993	Tuttobene	.....	221/7	5,902,027	A	5/1999	Robey et al.	
5,385,265	A		1/1995	Schlamp			5,915,909	A	6/1999	Smith	
5,386,462	A		1/1995	Schlamp			6,010,239	A	*	1/2000	Hardgrave et al. .... 364/487.01
5,389,919	A	*	2/1995	Warren et al.	.....	340/825.31	6,016,064	A	*	1/2000	Saeki ..... 235/375
5,467,892	A		11/1995	Schlamp			6,064,979	A	*	5/2000	Perkowski ..... 705/26
5,499,707	A	*	3/1996	Steury	.....	235/381	6,123,223	A	*	9/2000	Watkins ..... 221/121
5,544,996	A		8/1996	Castaldi et al.			6,259,367	B1	*	7/2001	Klein ..... 340/572.1
5,645,390	A	*	7/1997	Filberti et al.	.....	414/390	6,331,812	B1	*	12/2001	Dawalibi ..... 340/5.2
5,671,362	A	*	9/1997	Cowe et al.	.....	395/228	6,344,796	B1	*	2/2002	Ogilvie et al. .... 340/568.1
5,774,053	A		6/1998	Porter			6,356,941	B1	*	3/2002	Cohen ..... 709/219
5,820,237	A		10/1998	Robey			6,404,337	B1	*	6/2002	Van Till et al. .... 340/569
5,836,662	A		11/1998	Robey			6,456,900	B1	*	9/2002	Kakuta ..... 700/233
5,890,136	A	*	3/1999	Kipp	.....	705/22	2003/0052778	A1	*	3/2003	Wong ..... 340/540

\* cited by examiner



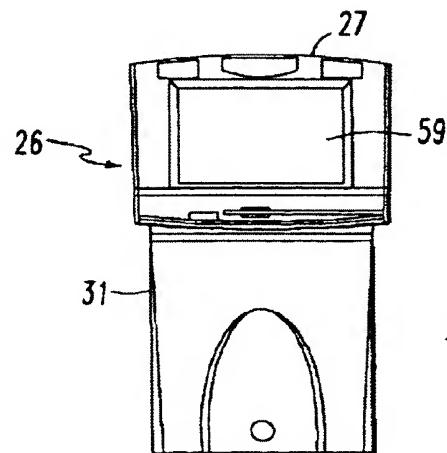
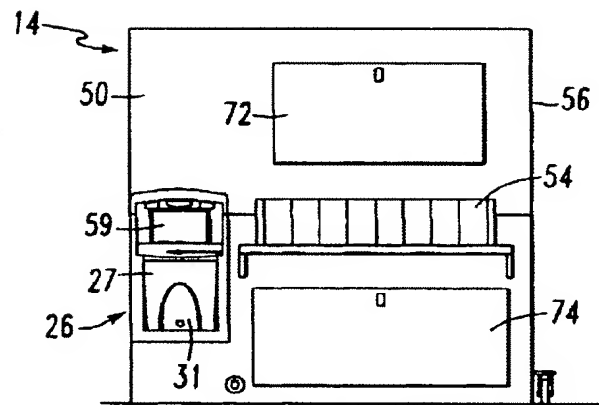
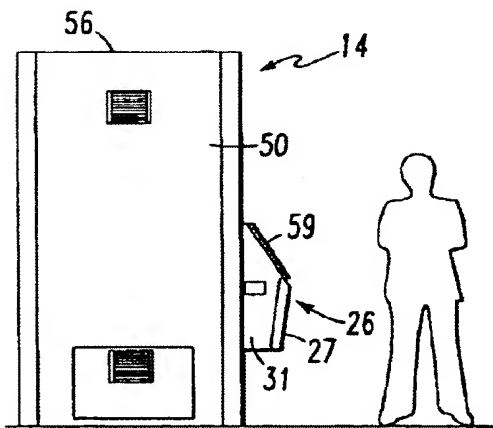
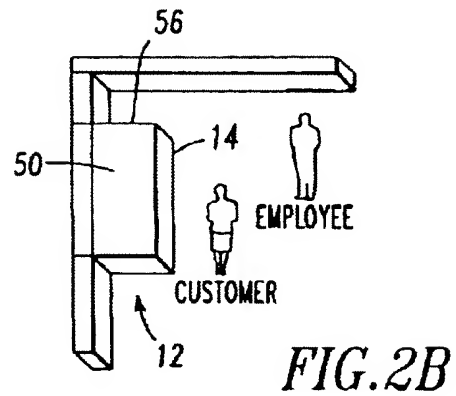
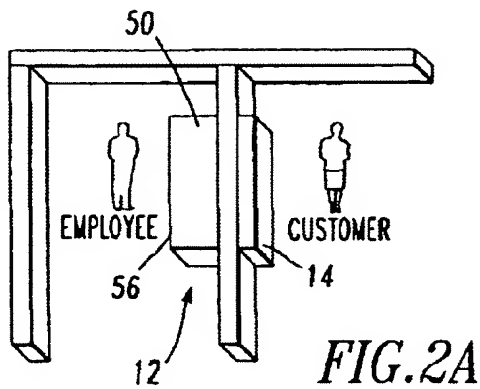


FIG. 5

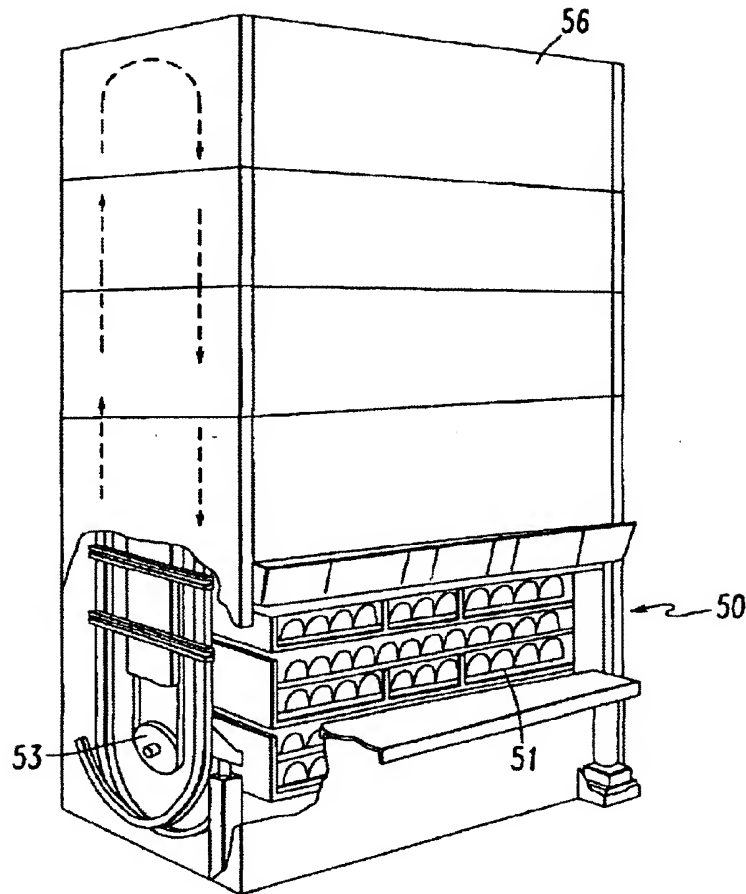
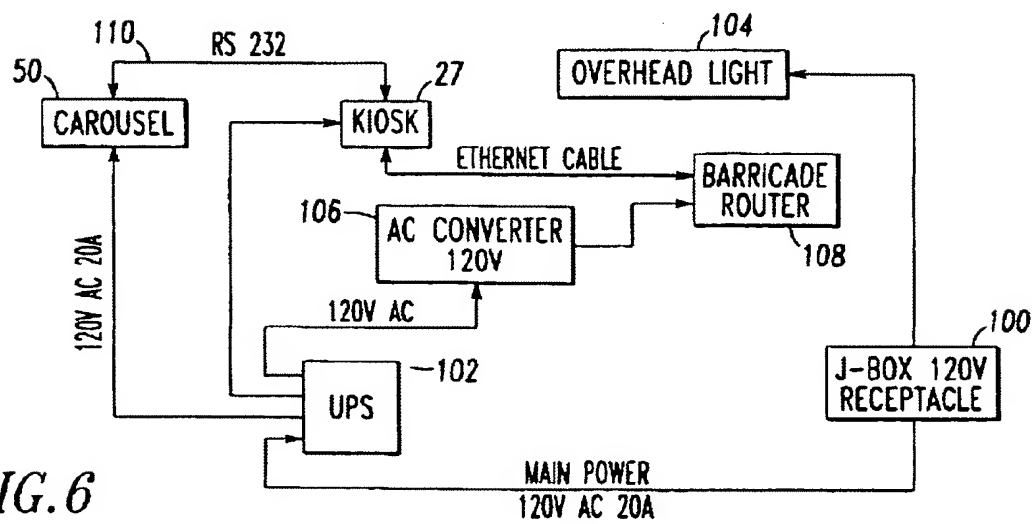


FIG. 6





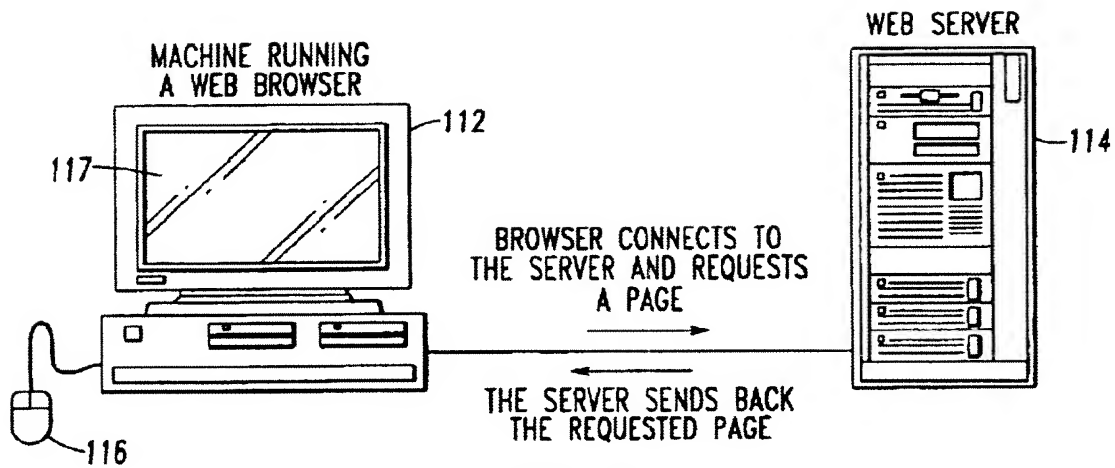


FIG. 7

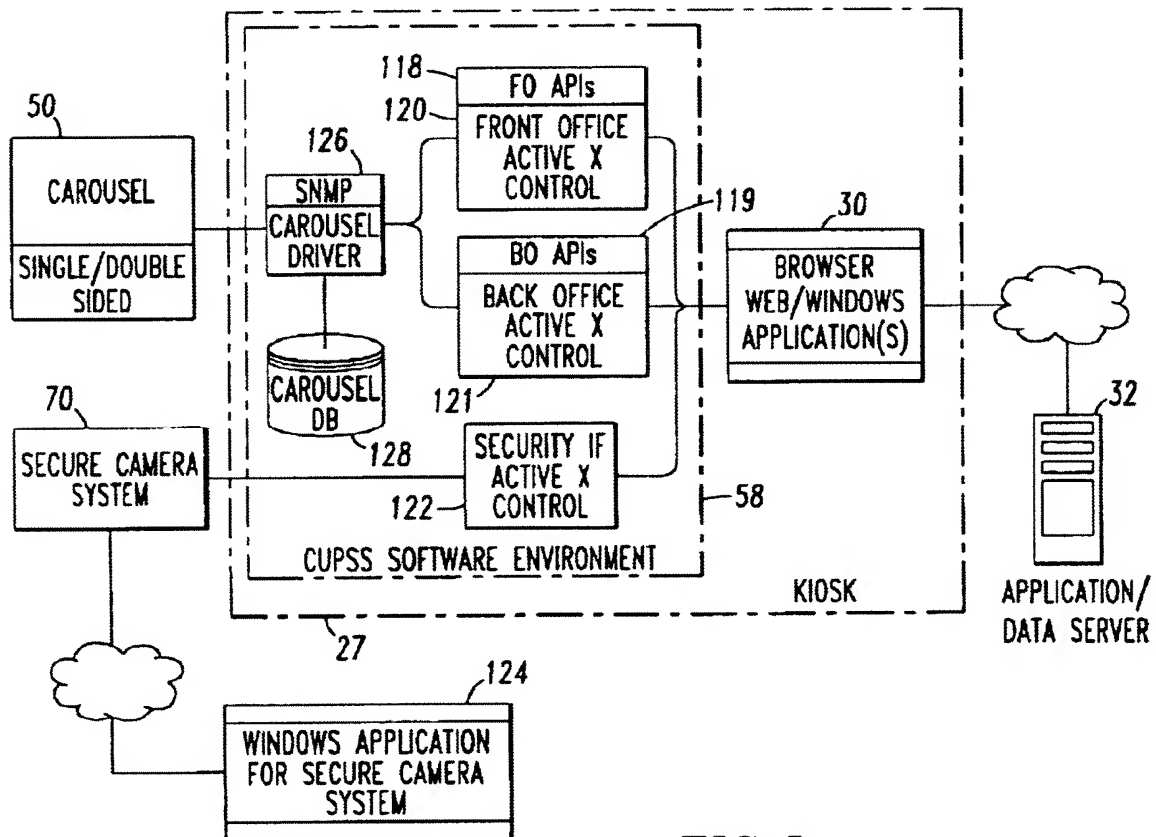


FIG. 8

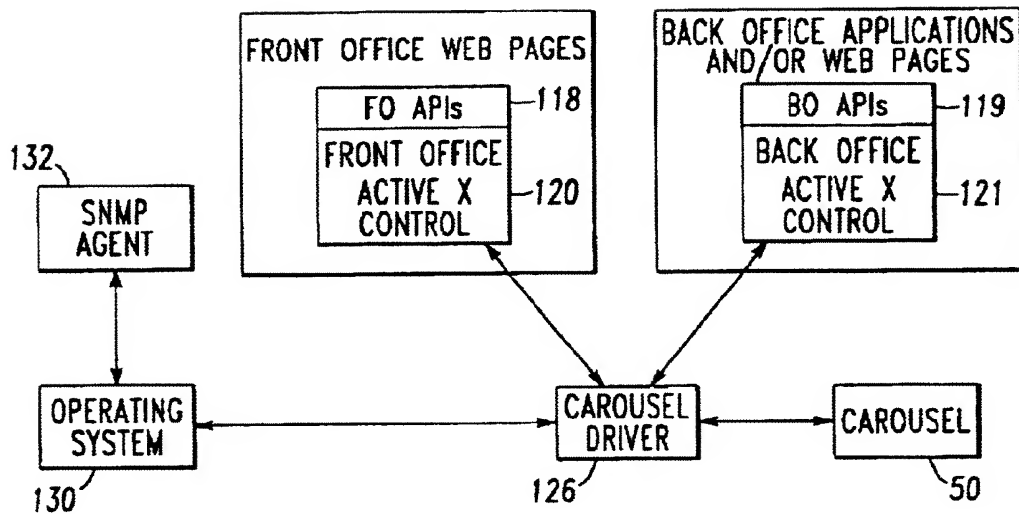


FIG. 9

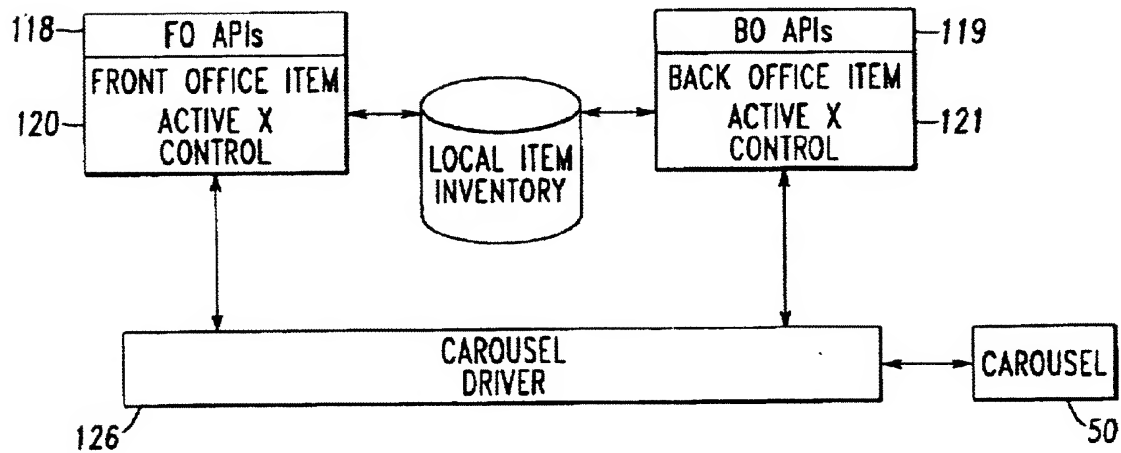


FIG. 10

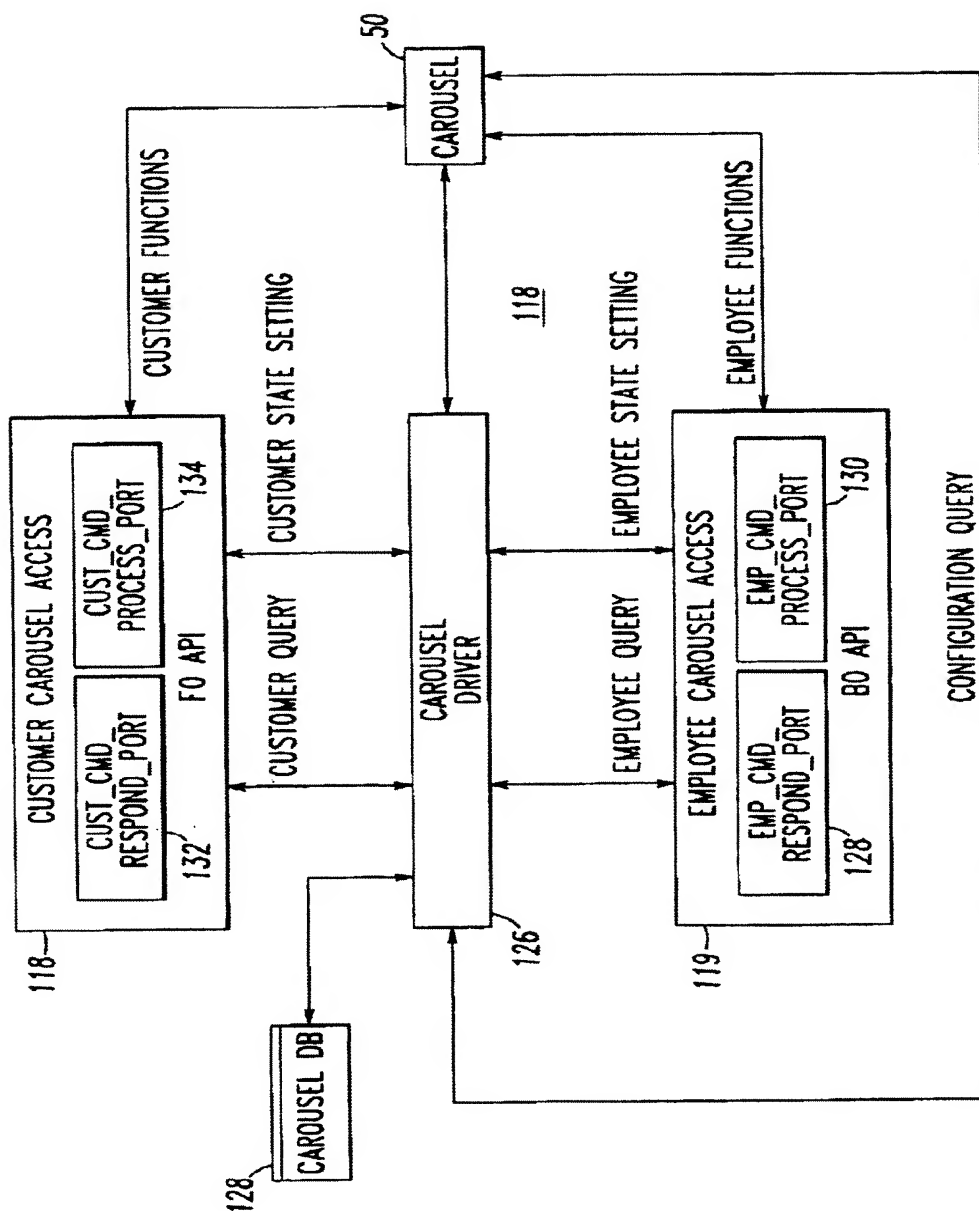


FIG. 11

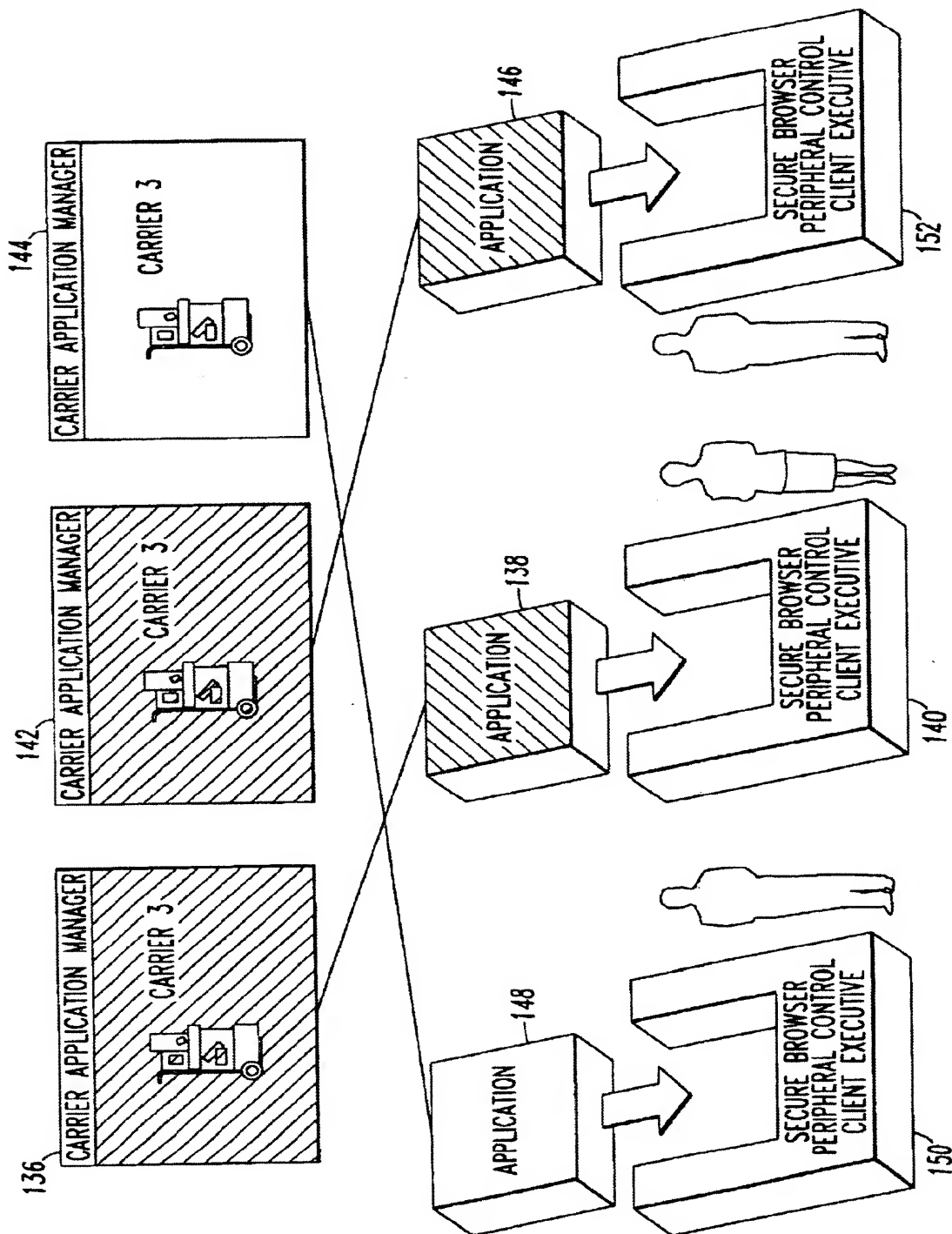


FIG. 12

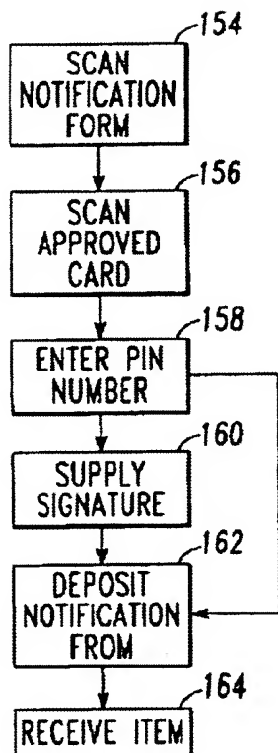


FIG. 13

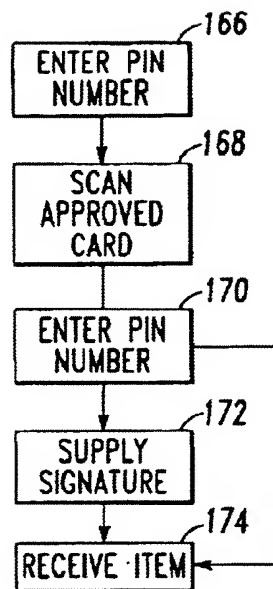


FIG. 14

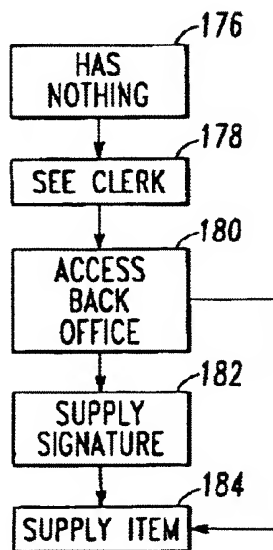


FIG. 15

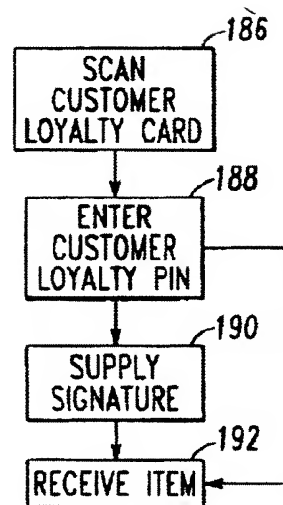


FIG. 16

1

**ITEM DELIVERY AND RETRIEVAL SYSTEM****ORIGIN OF THE INVENTION**

This application is a non-provisional application including the subject matter and claiming the priority dates of Provisional Application No. Serial No. 60/220,842, filed on Jul. 26, 2000 and Provisional Application Serial No. 60/265,875 filed on Feb. 5, 2001, the contents of which are meant to be incorporated herein by reference.

**BACKGROUND OF THE INVENTION**

This invention relates generally to item storage and retrieval systems and more particularly to a web-enabled item storage and retrieval system including a secure enclosure which is controlled by computer apparatus employing browser technology type software.

The overnight delivery business is a highly competitive business, requiring delivery companies to develop innovative approaches to reduce delivery cost and increase customer satisfaction. With today's lifestyles, persons, i.e., customers, are frequently not at home to accept deliveries and/or it is inconvenient to return items. Thus there is a need for eliminating the requirement of couriers, meaning persons employed by a delivery company to make a delivery to a customer, to make multiple visits to the same residence or small business in order to complete delivery transaction(s).

Accordingly, there is a need for a secure item and delivery and return system which permits a customer to retrieve undelivered items or return items at any hour of the day, seven days a week. Typically, a customer receives some type of notification that an undeliverable item is stored at a remote location where there is located an item delivery and retrieval system. When it is convenient, the customer subsequently travels to the location of the system and retrieves the items. The benefits of such a system include labor savings, increased customer satisfaction, improved traceability, and improved process control and item security.

**SUMMARY**

Accordingly, it is an object of the present invention to provide a method and apparatus for storing items of various types, sizes and shapes for subsequent retrieval or return when an initial delivery was unsuccessful.

It is a further object of the invention to provide an item delivery and retrieval system which is operable in multiple utilization scenarios.

It is yet another object of the invention to provide an item delivery and retrieval system which is accessible on demand by either delivery and/or storage clerks (employees), and clients (customers) wishing to store or retrieve undelivered items.

It is a further object of the invention to provide an item delivery and retrieval system which provides a requisite amount of security for items stored therein while providing relatively easy and user friendly access.

And it is still a further object of the invention to provide an item delivery and retrieval system which is controlled by application configurable digital computer apparatus supporting browser and web page software.

The foregoing and other objects are achieved by a storage subsystem and a computer subsystem. The storage subsystem provides a secure items storage and delivery environment including a secure enclosure having an item storage carousel including controller apparatus as well as a set of

2

sensors. The computer subsystem is embodied in web page based customized application software for implementing an application interface of selectively configurable application interface controls, such as ActiveX controls, for providing user access to one or more storage bins located behind a set of normally closed doors which are selectively opened and then closed for item storage and retrieval, provides access control to the bins, and manages the location of the items in the storage sub-system. The doors are opened when proper identification is provided by a user so as to permit access only to specifically designated bin(s).

Further scope of applicability of the present invention will become apparent from the detailed description provided hereinafter. It should be understood, however, that the detailed description and specific example, while disclosing the preferred embodiment of the invention, is given by way of illustration only, since various changes and modifications within the spirit and scope of the invention will become apparent to those skilled in the art from the following detailed description.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention will become more fully understood when the detailed description provided hereinbelow is considered together with the accompanying drawings which are provided by way of illustration only and are thus not meant to be limitative of the subject invention and wherein:

FIG. 1 is a block diagram broadly illustrative of the system architecture of an item delivery and retrieval system (IDRS) in accordance with the subject invention;

FIGS. 2A and 2B are illustrative of double sided and single sided item delivery and retrieval configurations of an IDRS in accordance with the subject invention;

FIGS. 3A, and 3B are illustrative of left side and front elevational views of a single sided vertical carousel assembly forming a part of the IDRS so as to provide a secure enclosure in accordance with the preferred embodiment of the subject invention;

FIG. 4 is a partially cutaway perspective view of the rear portion of the vertical carousel assembly shown in FIGS. 3A-3D.

FIG. 5 is a partially cutaway respective view of the rear portion of the vertical carousel assembly shown in FIGS. 3A-3D;

FIG. 6 is an electrical block diagram illustrative of the electrical system powering the apparatus in accordance with the subject invention;

FIG. 7 is a block diagram illustrative of how web servers operate to request and receive a web page;

FIG. 8 is a block diagram further illustrative of the system architecture of the IDRS in accordance with the subject invention;

FIG. 9 is a block diagram illustrative of the basic carousel control architecture of the subject invention;

FIG. 10 is a block diagram illustrative of the enhanced item control architecture of the subject invention;

FIG. 11 is a block diagram further illustrative of the carousel driver interface of the subject invention;

FIG. 12 is a block diagram illustrative of an application of the item delivery and retrieval system in accordance with the subject invention; and

FIGS. 13, 14, 15 and 16 are simplified flow charts illustrative of four modes of utility of the subject invention.

**DETAILED DESCRIPTION OF THE INVENTION**

Item delivery companies incur a high cost to make multiple deliveries at one location if a customer is not at

3

home. The high cost results from: redeliveries that increase the delivery expense through additional man-hours and use of valuable space on a delivery truck; deliveries left at the delivery point without any signature are subject to theft, damage and lack delivery verification; and there is no method to handle returns. The customer also has concerns about the deliveries, namely: redeliveries are inconvenient; deliveries are difficult to schedule and wait for re-delivery; there are concerns about theft and weather damage to packages; and returning is a time-consuming and often irritable task.

Furthermore, delivery companies are belabored with item process control, typically: significant labor hours to hand-write left notices, e.g., first delivery attempt, second notice attempt, or final notice prior to returning to sender; the lack of visibility of the item while in the on-delivery, re-delivery, or return to sender life-cycle; manual process generates significant hard copy content to manage, store, protect and archive; and, hard copies are cumbersome to obtain quick visibility.

In accordance with the problems briefly referred to above, this invention is directed to an item delivery and retrieval system (IDRS) which stores a variety of products and items from post cards to large packages. The system may be installed in three scenarios: (1) behind the customer service counter for operation by employees; (2) free standing in a public access location for access by both the employees or customers; or (3) wall mounted in a public location as a customer operated system. If wall mounted, the front of the IDRS is accessible by customers in a common area or lobby, while the rear of the IDRS is accessible by employees/clerks for behind the scenes loading of items.

The IDRS in accordance with this invention is comprised of a single sided or a double sided storage subsystem and a computer subsystem. The storage subsystem provides secure item storage and delivery. The computer subsystem includes separate customer and employee interfaces, provides access control, and manages the location of items in the storage subsystem.

When necessary, multiple IDRS(s) may be co-located at a single facility, allowing the delivery company to configure the system based on site requirements. Multiple IDRS systems can be integrated, when desirable, with multiple storage and computer subsystems for efficiently serving a higher volume of items and customers.

Referring now to the drawings wherein like reference numerals refer to like components throughout, FIG. 1 is broadly illustrative of the architecture for an IDRS system 10 including, among other things, a storage subsystem 12 and a computer subsystem embodied in a front office client module 14 and a back office module 16, both of which includes state of the art computer apparatus with application configurable software, such as a browser, which is internet web page based. These elements are interconnected by means of a local area network (LAN) 18 and a router/firewall 19.

As shown in FIG. 1, a master server 20 supports and stores set(s) of web pages. They are connected via a direct network connection 17 from a company wide area network 15 and connection 13 to user access terminals 24 and 26 supporting web browsers 28 and 30 located in the front office client module 14 and back office module 16.

Additionally, the master server 20 supports and stores set(s) of web pages that are connected via the internet 22 to a web server 32. The web server 32 is a pass through connection via the internet 22 to user access terminals 24

4

and 26 supporting web browsers 28 and 30 located in the front office client 14 and back office module 16. A modem 34 connects the user access terminals 24 and 26 to the web server 32. A modem 35 connects the master server 20 to the web server 32.

As illustrated, the front office browser software 30 and the back office browser software 28 reside in separate user access terminals 26 and 24. This would be the case for double sided load and retrieve system as shown in FIG. 2A; however, in a single sided system as shown in FIG. 2B, the front office browser software 30 and the back office browser software 28 would reside in a common terminal, i.e., the front office client terminal 26 which is in the form of a kiosk 27, shown in FIG. 4, and which is associated with the front office client module 14.

The web server 32 can also be internet connected to other software such as browsers 36, 38 and 40 located, for example, in another customer access terminal 42, a customer delivery terminal 44, or a personnel support terminal 46. The customer may view information about the items stored in the IDRS, for example, from terminal 42. This information may include date stored and type of item. The customer may also view any personalized information such as their e mail address and date of IDRS membership.

Delivery company personnel may view machine usage information such as is the IDRS full at certain locations and hardware failure information from a support terminal such as terminal 46 which is accessible by modem 45. The master server 20 is also shown connected to the delivery company-wide area network 15 which is coupled to the Internet 22 via a firewall 49 and connection 47.

The preferred embodiment of the storage subsystem 12 includes a vertical carousel 50, a single sided embodiment of which is shown in FIGS. 3A and 3B. The carousel 50 is constructed of individual carriers or shelves that travel on a chain and track as shown in FIG. 5. Vertical and horizontal mechanical inserts are mounted on the carriers with the insert determining the number of compartments associated with that carrier. The construction of the carriers and inserts preclude unauthorized access to adjacent compartments. The number and size of the compartments is furthermore configurable based on the delivery company requirements. The size of the compartment determines the size of the item which can be stored varying from postcard to large item. Each compartment is assigned a unique identifier identification number such as a sticker with a unique barcode for tracking items located therein. The computer subsystem keeps a database linking the storage compartment unique identifier with a unique mail piece identifier. A partially cutaway view of the single sided carousel structure is shown in FIG. 5 wherein a plurality of item holding trays 51 are moved up and down from front to back via a motor driven sprocket and chain assembly 53. This equipment is well known and comprises, for example, a vertical carousel manufactured and sold by Remstar International, Inc. of Westbrook, Me. Another known manufacturer is Hanel Storage Systems of Oakdale, Pa.

The carousel 50 also includes a set of sensors and a control system 52 (FIG. 1). The sensors allow the safe use of the storage subsystem by the general public. An optional safety light curtain is included across the customer access doors 54, as shown in FIG. 3B, to provide a means to stop the carousel or doors when obstructed by fingers, hands, arms or items. Internal sensors, not shown, detect items that obstruct the carousel's rotational flow. In the event of an obstruction, the motions of all access doors and the carousel

5

cease. Optional emergency stops, also not shown, are located on the periphery of the machine to allow an immediate stop of the machine. Setting of an emergency stop by delivery company personnel (employees) results in ceasing the motion of all access doors and the carousel. Additional sensors may be included in the vertical carousel to detect carousel movement and interface to external pushbuttons.

The carousel control system interfaces with the sensors and controls the movement of the carousel 50. The carousel control system responds to requests from the computer subsystem in either the back office module 16 or front office module 14 via a software carousel driver shown in FIGS. 8 and 9 and which will be considered subsequently. The carousel control system includes a diagnostic capability so as to provide diagnostic information regarding the safety light curtain, photoeyes, motor starters and external pushbuttons.

As shown in FIGS. 3A, 3B and 5, the carousel 50 is housed within a secure enclosure 56. The enclosure 56 is vandal resistant and graffiti resistant. The front doors 54 of the carousel 50 are segmented to allow the opening of a door in front of the desired compartment only. The height, width, depth of the enclosure is based on customer requirements and mechanical constraints.

The front office client module 14 provides a user friendly customer interface implemented in customized application software for the retrieval of an item. The term "application" is well known in the art and refers to a computer program for carrying out a certain function or producing a certain result. As shown in FIG. 1, the front office module 14 includes in addition to application configurable browser software 30 which resides in the user access terminal 26, a screen 59 which may optionally be a touch screen and other optional devices such as a barcode reader 60, credit/debit card reader 62, pin pad 64, receipt printer 66, signature pad 68, and two security cameras 70 and 71. While the front office client module 14 is preferably accessed from the front, it may be accessed from the front and/or rear depending on the customer requirements.

The front office user access terminal 26 is further shown in FIG. 4 consisting of a kiosk 27 having a touch activated screen 59 and a housing 31 wherein there is located the customized application software 58 for controlling the carousel 50.

The back office module 16 provides an interface also implemented in customized application software for employees to load the IDRS from front and/or rear access doors of the carousel 50. Two front access doors 72 and 74 are shown in the single sided carousel 50 shown in FIG. 3B. If the system does not require the carousel 50 to be rear loaded, the back office functions can be implemented on the customer interface side or front of the carousel 50 via the kiosk 27 as shown in FIG. 3B, but still may be accessed only by authorized delivery company personnel. In such a configuration, both software interfaces, i.e., a front office application program interface (FO API) and a back office application program interface (BO API) reside in the kiosk 27.

If the back office module 16 is located separate from the kiosk 27 such as where the carousel 50 is designed so as to be rear loaded from a back room, it would, for example, include a separate employee access terminal 24 equipped with its own application configuration browser software 28 as shown in FIG. 1. The terminal 24 would also include a screen 76 and other peripheral devices such as, but not limited to, a bar code reader 78, a modem 80 for connecting

6

to a bank clearing switch 82 and apparatus 84 for connection to an external telephone 86. Additionally, such a back office module 16 would include a printer 88 which is coupled to the local area network 18.

Also shown in FIG. 1 is a handheld wireless device/scanner 90 which can access the storage subsystem 12, the front office module 14 and the back office module 16 including a screen 91 via a wireless local area network (LAN) shown by reference numerals 92 and 94 which are coupled to the local area network 18 and allows for mobility of the handheld device/scanner 94. The handheld wireless device/scanner 90 may also execute an application to store items in the carousel 50 of the IDRS system 10.

It should be noted that a single back office module 16 can control multiple front office modules 14 and storage subsystems 12 at high demand sites. This feature allows the delivery company to vary the quantity of front office kiosks 27 and carousels 50 based on site-to-site variations on demand.

The master server 20 shown in FIG. 1 includes state of the art digital computer apparatus supporting master server application software and is used to network the subject system 10 as well as multiple other systems together over the delivery company wide area network 15. The Master Server 20 allows delivery company supervisors and operations managers to browse any website(s) to determine usage rates across sites and system availability information. The master server 20 contains the centralized data for the IDRS system such as certain data indicating IDRS locations, user e-mail addresses, user account/loyalty card information, item status, and any other information needed to operate the system. Other master servers, not shown, may be linked to geographic regions for large or regional deployments. Customers may access the specific website to get item delivery traceability information. The firewall 49 prevents the public from corrupting the Master Server data and ensures data integrity.

Referring now briefly to FIG. 6, shown thereat is an electrical block diagram of the electrical power supplied to the equipment shown in FIGS. 3A, 3B, 4 and 5. 120 VAC electrical power is fed from an outside power line to a junction box/receptacle 100 where it then is fed to an AC power supply 102 and an overhead light 104. The power supply 102 feeds AC power on separate busses to the carousel 50, the kiosk 27 and a 120V AC converter 106 in a conventional manner. The output of the converter 106 is fed to a router 108 which provides an internet cable connection to the kiosk 40. An RS 232 communication cable 110 is shown connected between the carousel 50 and the kiosk 27.

Before considering the details of the application software of this invention, reference is first made to FIG. 7 which is intended as a simple tutorial to illustrate how web browser technology is utilized to display a web page. As is well known, a web browser is a software application used to locate and display a web page, i.e., a document on the World Wide Web. As shown, reference numeral 112 denotes a machine running web browser software connected to a web server 114. Reference numeral 116 denotes a mouse, i.e., a well known hand activated device to move a cursor on a computer screen or activate a command, connected to the machine 112. Thus when a web page is desired, the browser software in the machine 112 connects to the server software in the web server 114 and requests a page. The web server 114 in turn retrieves the requested page from a digital storage located, for example, in a master server 18 shown in



FIG. 1, where it is then sent back to the machine 312 running the web browser where it is then displayed on a screen 117.

Referring now to FIG. 8, shown thereat is a simplified block diagram of the subject invention and illustrative of the software architecture in accordance with the preferred embodiment of the invention where the front office application program interface (FO API) 118 and the back office application program interface (BO API) 119, referred to above, are located in the CUPSS software environment 58 of the kiosk 27 (FIG. 4) using ActiveX control technology. As shown, the FO API 118 and BO API 119 support ActiveX controls 120 and 121. A security interface is also shown using ActiveX and control 122.

ActiveX control is a well known concept in current state of the art of digital computer technology. It is a programming language including a set of rules for how applications should share information and can be automatically downloaded and executed, for example, by a web browser. ActiveX controls have full access to a windows operating system using web pages. ActiveX control is particularly adapted to implement custom controls, which in the subject invention comprises the FO API 118, the BO API 119 and a carousel driver 126 which is connected to the carousel controller 38 (FIG. 1).

The FO API 118, the BO API 119, and the carousel driver 126 combine together to form a customized application and carousel independent interface which is configured on demand to meet a desired configuration of utilization. Accordingly, the carousel driver 126 can be instantaneously used to control any manufacturer's carousel simply by enabling the particular manufacture software switch and recompiling the driver associated therewith.

The configuration of the carousel 50, e.g. bin locations and size, is controlled by a carousel database 128 also residing in the CUPSS software environment 58. The carousel driver 126 supports both double sided and single sided configurations such as shown in FIGS. 2A and 2B. The carousel driver 126 coordinates access to the carousel 50 such that only one employee or customer operates the carousel at one time. For employee access, the carousel driver 126 opens front and/or rear doors, e.g. doors 72 and 74 shown in FIG. 3B, exposing multiple compartments authorized to be accessed by the employee. For customer access, the carousel driver 126 opens the front doors 54, exposing a single compartment authorized to be accessed by the customer.

The carousel driver 126 also interacts with an operating system 130 and a simple network management protocol (SNMP) agent 132 as shown in FIG. 9 to ensure a safe environment is maintained during storage personnel/employee or customer/client operation. Status information from light curtains, door movement, carousel movement, and power fluctuations is constantly maintained. The carousel driver 126 uses the information to control the load and retrieval process so that the integrity of the carousel 50 is maintained, such as closing the doors during a power failure, and the safety of the user is maintained just closing the door while the user is reaching into a bin.

FIG. 9 is further illustrative of the control interface which controls the carousel 50 by way of the carousel driver 126 to rotate the carousel and to open and close doors and then completely manages any items that go into and out of the carousel. The ActiveX controls 120 and 121 are furthermore active only for the processing time of the applications or web pages that contain them. The major function of the ActiveX controls 120 and 121 in basic carousel control architecture

shown in FIG. 9 can be summarized in the following table I.

TABLE I

Front Office Control Functions	Back Office Control Functions
Connect	Connect
Cue Bin Location	Open All Doors
Open Bin Location	Open Bin Location
Close Bin	Rotate Carousel
	Identify Bin
	Close Bin
	Close All Doors

The Connect function initializes connections of the ActiveX controls 120 and 121 to the carousel driver 126. The ActiveX control may also be required to pass an identification code to the carousel driver 126 for access control security. The Cue Bin Location function is used by the FO API 118 to rotate the carousel 50 such that the requested bin is positioned behind the doors 54 without any of the doors being opened. This function is used to reduce the service time required for the overall transactional session, if the operational rules of the application also include authentication of the user. The Cue Bin Location function will position the carousel 50 while the transactional process of authenticating the user takes place. This will reduce the overall transaction time. The Open Bin Location function is used by the BO API 119 and FO API 118 to position the carousel 50 and to open the doors to a specified bin. This may require an access code. The Open Doors function is a back office function that is used to gain full access to the carousel 50. This function may restrict access based on identification code. The Rotate Carousel function is used by the BO API 119 to position hidden carriers to the access point and may restrict access based on identification code. The Identify Bin function is used by the BO API 119 to identify a particular bin when all doors are open. This function may be used by applications to verify if bins are empty or indicate which items need attention. The Close Bin function is used by the FO API 118 and/or BO API 119 to close the doors. Once the door has been opened, the Close Bin function may also be used to clear bin access codes. The Close All Doors function is used by the BO API 119 to close all doors and secure the carousel 50.

The present invention also contemplates an enhanced item controlled architecture shown in FIG. 10 which provides an interface to applications via ActiveX controls 120 and 121 for providing, among other things, inventory control of items that are placed into or out of the carousel 50. This enhanced architecture provides advanced functionality and allows multiple delivery companies to use a single IDRS carousel 50. This interface is more transactional based and permits an application to load items, find empty locations, remove items and a host of transactional type of information queries. Again, the carousel driver 126 is a persistent service of the operating system and the ActiveX controls are active only for the processing time of the applications or web pages that contain them. The enhanced architecture additionally includes a local item inventory database 134, but uses the same interfaces 120 and 121 to the carousel driver 126 for carousel control, but provides a higher level of service to the application through its APIs 118 and 119. Access codes that are required by the carousel driver 126 and are not provided by the application are generated by the ActiveX controls 120 and 121.

Application access for the enhanced item controlled architecture to the functions to be described can be classified in

9

two types of control classes: (a) session access and, (b) bin access. Session access describes the protocol required to any given application to connect to the carousel driver 126. Bin access describes the protocol for a qualified application to reserve or lock any given bin.

Session access is controlled by means of an access control list (ACL) which is maintained in the data of the carousel driver 126. As is well known, a "list" is an ordered set of data which is normally accessed in a digital computer sequentially. The ACLs of the FO API 118 and BO API 119 will contain the ACL member ID of all authorized applications of the carousel 50. When an application initializes its embedded ActiveX controls 120 or 121, it in turn establishes the requisite transmission control protocol (TCP) connections to the carousel driver 126. The ACL member ID that is passed with the connection request will be checked against the carousel's ACL. A successful match will permit the connections to be made, assuming no other connection is established. An unsuccessful match will reject the connection and not permit that application to have access to the carousel 50. If there are no members in either ACLs, then it should be assumed that any application can access the carousel and no access security will be required to operate the carousel.

With respect to bin access, the carousel driver 126 will grant access to any given bin based on the access type declared for that bin at installation time. Each bin will be set up based on one of two access types Static or Dynamic.

The Static access type relates a given bin to a given application on the ACL. This type of bin access petitions the carousel 50 to either a single application or multiple applications with fixed storage capabilities. The Dynamic bin access type allows for more efficient use of the carousel 50 in the multiuse configuration by allowing applications to gain access to the bins based on a common pool of dynamically allocated bins. Once a bin has been accessed, the application may place or remove a lock on that bin with an application supplied access code. Subsequent access to that bin or removal of the lock will then require the access code for that bin. The carousel driver 126 will journal log all access activity via a simple network management protocol (SNMP). This information will provide the basis for "use accountability" for owners/administrators of the equipment.

It should be noted that if more than one member exists in the ACL of the BO API 121, back office operations will limit exposure of the bins, i.e., rotation operations, to only those bins which have any given application is authorized to use. This may be accomplished by closing all doors before a rotation and only granting open doors at authorized carrier level as will be described subsequently with respect to FIG. 12.

The Static bin access type is the simpler of the two access services. The configuration of the carousel 50 is segmented into a predetermined configuration which specifies who has the right to access any given bin. If no ACL member is specified, it would be assumed that any application has access to the bin. At configuration time, it should be noted that the segmentation definition will take into account for the dual sided and/or single sided system as shown, for example, in FIGS. 2A and 2B such that unauthorized bins will not be exposed during back office operations.

The Dynamic bin access has two modes of operation, with or without back office operations. Dynamic bin access without back office operations will permit any application to access any unlocked bin. Once the bin has been locked with an access code, both the ACL member ID and access code will be needed to re-access the bin or remove the lock.

10

Dynamic bin access with back office operations, however, will operate as above, but with a further restriction such as to limit access to those bins where no other bin on that carrier, for single sided configurations and adjacent carrier for dual sided configurations, is locked by another ACL member ID.

The major function of these ActiveX controls for the enhanced architecture shown in FIG. 10 are summarized in the following Table II.

TABLE II

Front Office Item Functions	Back Office Item Functions
Connect	Connect
Cue Item/Authenticate User	Register Item
Load Item	Purge Item
Remove Item	Load Item
Close Bin	Remove Item
Return Item	Open All Doors
Query Item	Open Bin Location
Print Receipt	Identify Bin
	Rotate Carousel
	Close Bin
	Close All Doors
	Database Maintenance/Reports

With respect to the functions listed in Table II, the Connect function, for example, initializes connections of the ActiveX control of FO API 118 and BO API 119 to the carousel driver 126. The ActiveX controls may also be required to pass an identification code to the carousel driver 126 for access control security. This function is the same as in the basic control outlined in Table I. The Cue Item function is similar to the Cue Bin Location function of Table I and is used by the FO API 118 to rotate the carousel 50 such that the requested item is positioned behind the doors 54 without any of the doors being opened. This function is also used to reduce the service time required for the overall transactional session. If the operational rules of the application include authentication of the user, the Cue Item function will position the carousel 50 while the transactional process of authenticating the user can take place, and thus will also reduce overall transaction time. The Register Item function is used by the BO API 119 to register an item and the item characteristics in the inventory data base 134 (FIG. 10). This function may be used to set the bin access code and may use an external scanner or similar data entry device. The Load Item function is similar to the Open Bin Location function (Table I) and is the function used by both the BO API 119 and the FO API 118 to position the carousel 50 and open the doors, for example, 72 and/or 74 of FIG. 3B for a specified item at a specific location. The item is then registered in the local database 134. This function may also be used to set the bin access code and may use an external scanner or similar data entry device.

The Purge Item function is used by the BO API 119 to remove an item in the local data base 134 and clear the bin access code. This function may require a bin-access code and also may use an external scanner or similar data entry device. The Close Bin function is used by FO API 118 and/or BO API 119 to close the doors 54, 72, 74. The Remove Item function is similar to the Open Bin Location function of Table I and is the function used by both the BO API 119 and the FO API 118 to position the carousel 50 and open the doors 54 to a specified item. The item is then marked as removed from the local database 134 and the bin access code is cleared if a bin access code is present.

The Return Item function is used by the FO API 118 to close the bin doors 54 and flag/mark the item in the database

11

134 for return. This function may also be used to flag an item that has not been removed from the carousel 50 but has been purged from the database 134. This function may be used to set the bin access code and is similar to the Remove Item and the Load Item function, noted above, with an item that is already in the system. The Query Item function is used by the FO API 118 to find and load time and status information into the database 134 regarding item removal or return. The Print Receipt function is used by the FO API 118 to print a transaction receipt of item removal or return from the carousel 50.

The Open All Doors function is a function of the BO API 119 that is used to gain full access to the carousel 50. The Open All Doors function may restrict access based on an identification code and is the same as in the basic control outlined in Table I. The Open Bin Location function is used by the BO API 119 to position the carousel 50 and to open the doors 72 or 74 to a specified bin and may require an access code. Again, this function is the same as in the basic control outlined above with respect to Table I. The Identify Bin function is used by the BO API 119 to identify a particular bin when all doors are opened. This function may be used by applications to verify if bins are empty or indicate which items need attention. This function is also the same as in the basic control outlined above.

The Rotate Carousel function is used by the BO API 119 to position hidden carriers to a specific access point and may restrict access based on an identification code. This function is also the same as in the basic control. The Close All Doors function is used by the BO API 119 to close all doors and secure the machine and is the same as in the basic control described with respect to FIG. 9. Finally, the Database Maintenance/Reports function is used by the BO API 119 to update the database 134.

Other queries and maintenance functions of the local item inventory base will depend on the design of the database itself.

With respect to the three major interfaces considered above with respect to FIGS. 8, 9 and 10, namely: the employee or BO API 119; the customer or FO API 118, and the carousel driver interface 136, the employee or BO API 119 has access to the carousel driver 126 as shown, for example, in FIG. 11 through an immediate response port termed a "command respond port" 128 or a process generate event port termed a "command process port" 130. The command respond port 128 will return with the function result. The command process port 130 will return the success of sending the message upon receiving the completion or error of a command. This port will generate an event with the status of the last command. The attached Appendix A is illustrative of the set of functions implemented by the employee interface or BO API 119.

The customer or FO API interface 118 has access to the carousel driver 126 through an immediate response port termed a "command respond port" shown by reference numeral 132 or a process and generate event port termed a "command process port" 134 shown in FIG. 11. The command respond port 132 will return with the function result. The command process port 134 will return the success of sending the message and upon receiving the completion or error of a command, this port will generate an event with the status of the last command. The attached Appendix B is illustrative of the set of functions implemented by the customer interface or FO API 118.

As noted above, the carousel driver interface 136 is an executable program that communicates directly with the

12

carousel 50, with both the customer FO API 118 and employee BO API 119. ActiveX controls 120 and 121 communicate with the carousel through this driver. The attached Appendix C is illustrative of the set of functions implemented by the carousel driver interface 126.

It should be noted that ActiveX controls can be used, without modification, by any development environment such as the Web. The application programming interface (API) remains constant, irrespective of whether a web page of a windows application is operating the carousel 50. This significantly reduces the software effort because the same API is used in both the Web and programming development environments. In addition, by hiding the peripheral details, this common use interface provides higher level interfaces to the developers, resulting in shorter time-to-market efforts.

For example, FIG. 12 is illustrative of a multiple user scenario. In FIG. 12, carriers refer to delivery companies. Accordingly, when a user approaches the IDRS system 10, he/she enters which item(s) they wish to retrieve, for example, using the kiosk 27. If delivery company 1 shown by reference numeral 136 delivered the item(s) to be retrieved, then delivery company 1's application 138 is plugged into the browser peripheral control portion 140 of the FO API 118 and executed by the Front Office ActiveX control 120 shown, for example, in FIGS. 8-10. At this time, delivery company 1 has control of the carousel 50 and can only access the designated items. The carousel driver 126 prevents any access to any other delivery companies, items or information. After the user has completed the transaction, all information with respect to the user, the delivery company and transaction is flushed from the carousel database 128. Thus a virtual architecture is generated which allows each delivery company, for example, delivery companies 2 and 3 designated by reference numerals 138 and 140 to function with confidence so that no other delivery company can view or gather any of its private information. As shown in FIG. 12, the delivery companies 2 and 3 can insert their respective applications 146 and 148 to respective browser peripheral control portions 150 and 152, which would then be executed in turn.

Considering now FIGS. 13-16, shown thereat are four step sequences outlining four possible modes of operation. Typically, a user, e.g., an employee of a delivery service company operates the IDRS in accordance with the subject invention from behind a customer service counter. A second user, e.g., a customer of the delivery service company interfaces with the IDRS system 10 using the front office client module 14 and retrieves the items from the storage subsystem module 12. Four scenarios are provided for customers to retrieve undelivered items, namely: (1) bar-coded notification form; (2) internet e-mail notification; (3) customer loyalty card (similar to supermarket savings cards and library cards with a magnetic strip on the back); and (4) front counter clerk.

The notification form approach (1) requires the delivery company courier to leave a written notice at the residence or business of attempted delivery. The written notice has a barcode on the form matching a self-stick barcode label placed on the item. When the delivery of an item cannot be completed, the courier will fill out a notification form, peel off a self-stick barcode label, and apply it to the item. The form is left at the address and the item is brought back to the IDRS 10. Once back at the delivery facility, the employee uses the back office subsystem module to initiate loading the storage unit 12 including the carousel 50. The screen on the terminal 28 in the back office subsystem module 16 displays the available compartments in the carousel 50. The

13

employee then selects an empty compartment to match the item size. The application software in the back office subsystem module 16 automatically requests the carousel 50 to move the compartment to the loading position and the doors of the carousel are opened. The employee scans the self-stick barcode label and an IDRS storage location barcode label is scanned and fed into a database.

Thereafter, a customer retrieves the items via the notification form. As shown in FIG. 13, at step 154, the customer scans the barcode on the notification form into the system at the kiosk 27 using the barcode reader 60 (FIG. 1). The IDRS ActiveX software described above uses the scanned barcode to reference the proper storage location linked to the form's barcode. Thereafter, an approved card provided by the delivery company for delivery authentication is scanned at step 156. If the delivery company requires, the customer uses a credit card, debit or customer loyalty card to authenticate the identity of the customer. Payment may be accepted for the transaction if the delivery company requests payment. A PIN number associated with the card is entered per step 158. This information is remotely verified and authenticates the user so that the card holder information tracks the person who picked up the item. The customer will then be prompted to supply a signature in accordance with step 160 via the signature pad 68 or on a touch screen 59 of the kiosk 27. This signature also tracks the person who signed for the item. Thereafter, the doors 54 of the carousel 50 automatically opens to the storage location of the customer's item. The customer then is prompted to deposit the notification form per step 169 into a slot and the previously undelivered item is retrieved per step 164. During this process, photos of the person retrieving the item may also be required using the cameras 64 shown in FIG. 1.

The second scenario involves internet e-mail notification (2). This approach requires notifying the customer via a supplied e-mail address, contained in a database of the master server 20 whenever an item is stored in the IDRS. In such an operational mode, the customer is first registered for service via the Internet by accessing a website and requesting internet e-mail notification service. At a minimum, a delivery address is provided to re-direct to the IDRS system. An e-mail address is provided to receive the notification. After registering, the customer must activate the service by calling the IDRS from a phone at the address given during registration. A customer selects a delivery profile, e.g., automatic placement of the item in the IDRS system 10. The customer indicates a preference to automatically put deliveries into the carousel 50 and thereafter eliminate any further attempts to deliver to the customer's address.

When an item is then stored in the carousel 50, an e-mail is sent to the e-mail address on file. The e-mail contains instructions on how to retrieve the item, including a six-digit PIN along with the location of the IDRS system, i.e., the address at which the IDRS 10 is located and, when desirable, with an optional map showing street locations, etc.

Items for the customer will be directed immediately to the IDRS 10 if the customer selected this delivery profile for this account. Not delivering the item reduces courier delivery time, delivery vehicle wear, and delivery vehicle gas and maintenance. The item may contain other delivery company barcodes such as expedite shipment confirmation of delivery, insured item, and indication of any other special handling. Any of these additional barcodes will also be scanned into the IDRS when the item is stored in the carousel. An e-mail is thereafter sent to the e-mail address on file associated with the item's delivery address.

As shown in FIG. 14, a customer would then go to the IDRS 10 and enter the 6-digit e-mail PIN on the PIN pad 64

14

as indicated by step 166. Next, a photo of the customer is taken via the cameras 70 shown in FIG. 1, whereupon the IDRS system 10 uses the e-mail PIN to reference the storage location(s) linked to the PIN. Next, the customer uses a card approved by the delivery company for delivery authentication. If the delivery company requires, the customer uses a credit card, debit or customer loyalty card to authenticate the identity of the user. Payment may be accepted for the transaction, if the delivery company requires payment. Next, the card is scanned via the card reader 62 in accordance with step 168 and the customer enters the PIN associated with the card. This is indicated by step 170. The information on the card is remotely verified and authenticates the user. If the delivery company requires, the IDRS 10 system will prompt the customer to supply a signature per step 172 via the electronic signature pad 68 or on the touch screen 59 (FIG. 5). Thereafter, the IDRS opens automatically to the store location of the stored item. The item is then removed from the storage location per step 174 and if the delivery company requires, a second photo of the item removal process is made.

The third scenario (3) is shown in FIG. 15 and one where a front counter clerk provides the necessary access information when a customer has lost or forgotten, for example, the notification form, e-mail/PIN or customer loyalty card/PIN or simply needs assistance at the IDRS 10 following storage of an item in the carousel 50 and where the customer had previously been alerted either by notification form or e-mail.

In such an instance, where the customer needs assistance as indicated by step 176, he/she would proceed to the front counter and see the clerk/employee per step 178 who would obtain the necessary information such as the delivery address and name and the necessary customer identification. The clerk then enters the address into the IDRS in the back office module 16 in accordance with step 180, whereupon the IDRS 10 uses the address to reference the storage location(s) linked to the address. The clerk/employee then retrieves the item(s) and upon receiving a customer signature per step 182, the item is supplied in accordance with step 184.

The fourth scenario (4) permits the customer to use a delivery company issued customer loyalty card to retrieve items stored in the IDRS. In this mode of operation, the customer would again register for service via the web by accessing a website and requesting customer loyalty service. This would again involve providing a delivery address to re-direct to the IDRS and an e-mail address to receive the notification. After registration, the delivery company mails a customer loyalty card to the customer.

Thereafter, the customer must activate the service by calling the IDRS from a phone at the address given during registration. The customer would then select a delivery profile, whereupon an e-mail notification is sent by the IDRS to the e-mail address on file. Contained in the e-mail are instructions on how to retrieve the item; however, there is no 6-digit PIN. Contained on the customer loyalty card is an encoded loyalty PIN number. The customer must then supply an associated PIN for authentication when using the customer loyalty card to access the IDRS.

Items will be directed immediately to the IDRS if a customer selected such a delivery profile for their account. The item may contain other delivery company barcodes such as expedited shipment confirmation of delivery, insured item indication of any other special handling required. Any of these additional barcodes will be scanned into the IDRS

15

when the item stored upon non-delivery. An e-mail is sent to the e-mail address on file associated with the item delivery address.

When the customer arrives at the IDRS, he/she enters the customer loyalty card and PIN via the card reader in the PIN pad as shown by steps 186 and 188 in FIG. 16. The cameras 64 would also take a photo of the customer. The IDRS system uses the customer loyalty account number to reference the storage location(s) of all items linked to the account. Authentication when necessary via signature is provided by the supply of a signature which would be prompted by the system per step 190. The doors 54 of the carousel 50 open automatically to the storage location of the item which is retrieved per step 192. Again, if the delivery

16

company requires, a second photo of the item removal process is taken via the cameras 64 shown in FIG. 1.

It should be noted that the flexibility of the IDRS system 10 in accordance with the subject invention allows the delivery company to deploy the appropriate configuration depending upon available floor space, item mix and capacity.

Having thus shown and described what is at present considered to be the preferred embodiment of the invention, it should be noted that the foregoing detailed description merely illustrates principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise various arrangements which although not explicitly described or shown herein, embody the principles of the invention and are thus within its spirit and scope.

## Employee Interface API

### Overview

The Employee Interface is the set of functions used by a NG application to provide a carousel service to an employee. The Employee Interface has access to the Carousel Driver through an immediate response port (command respond port) or a process and generate event port (command process port). The command respond port will return with the function result. The command process port will return the success of sending the message upon receiving the completion or error of a command, this port will generate an event with the status of the last command.

Events will be returned as one of the format: MsgEvent(INT msgVal, STRING tag)

Current values are as follows:

- -2 - Socket Error (use GetErrDesc() for description)
- -1 - Socket Error (not connected to Carousel Driver)
- 0 - OK: Operation complete
- 1 - ERR: error encountered while executing command (use GetErrDesc() for description)
- 2 - TIMEOUT: Result not received by the Driver within time period
- 3 - INUSE: Driver in use by the customer
- 4 - BUSY: Carousel Driver is busy processing another command
- 5 - HWERR: Hardware error that reports carousel errors

Registry values (in HKEY\_CLASSES\_ROOT\CUPSS):

- |                      |  |
|----------------------|--|
| CMD_PROCESS_TIMEOUT  | - Time period to wait for result from driver before returning "timed out" error. |
|                      | - If "0", then timeout is not used.  |
| CMD_RESPOND_TIMEOUT  | - Time period to wait for result from driver before returning "timed out" error. |
|                      | - If "0", then timeout is not used.  |
| DRIVER_HOST_NAME     | - Host name where driver is running (i.e. "127. 0. 0. 0" or "localhost")         |
| EMP_CMD_PROCESS_PORT | - Port number of the command process port (i.e. 5008)                            |
| EMP_CMD_RESPOND_PORT | - Port number of the command respond port (i.e. 5010)                            |

### Carousel Functions:

SHORT CloseDoors()

The carousel will close the doors that are currently open.

Return Format:

- 0 - Successfully sent information to Carousel Driver
- 1 - Socket error (call GetErrDesc() to get description of error)

STRING GetBinInfo(STRING binID)

Returns the carrier number, height, width, depth, and unit number of the specified location (binID).

Parameter Format:

BinID = U000000B00000

Return Format:

Unit:xx, Carrier:yy, Shelf:ss, H:hh, W:ww, D:dd

Where  
 xx = Unit Number  
 yy = Carrier Number  
 ss = Shelf Number  
 hh = height (inches)  
 ww = width (inches)  
 dd = depth (inches)

STRING GetCtrlVersion()

Returns the current version of the ActiveX control.

Return Format:

x.x (i.e. "0.9")

STRING GetDrvVersion()

Returns the current version of the carousel driver.

Return Format:

x.x (i.e. "0.9")

## APPENDIX A

2/3

**STRING**      **GetErrDesc()**  
 Returns an error description of the last function.  
 Return Format:  
     Error description (i.e. "Carousel Busy", "In Use by Customer")

**SHORT**      **GetMaxBins (STRING carrier, STRING shelf)**  
 If "carrier" and "shelf" are empty, then this method returns the maximum number of bins in current carousel configuration. Otherwise, the number of bins for the specified carrier and shelf are returned.  
 Return Format:  
     Max bins in carousel (i.e. 114)  
     OR  
     Max bins per carrier and shelf (i.e. 7)

**SHORT**      **GetMaxCarriers ()**  
 Returns the maximum number of carriers in current carousel configuration.  
 Return Format:  
     Max carriers (i.e. 12)

**SHORT**      **GetMaxShelves (STRING carrier)**  
 If "carrier" is empty, then this method returns the maximum number of shelves in current carousel configuration. Otherwise, the number of shelves on the specified carrier is returned.  
 Return Format:  
     Max shelves (i.e. 36)  
     OR  
     Max shelves per carrier (i.e. 3)

**STRING**      **GetState()**  
 Returns the current state of the customer access to the carousel.  
 Return Format:  
     {busy, error, idle, inuse, ready}

**SHORT**      **Initialize()**  
 Initialize the socket connections using registry values EMP\_CMD\_PROCESS\_PORT, EMP\_CMD\_RESPOND\_PORT and DRIVER\_HOST\_NAME.  
 Return Format:  
     Bit map of errors  
     1 - process port invalid  
     2 - respond port invalid  
     3 - no host name for driver

**SHORT**      **OpenAllDoors()**  
 The carousel will open all doors for employee access.  
 Return Format:  
     0 - Successfully sent information to Carousel Driver  
     -1 - Socket error (call GetErrDesc() to get description of error)

## APPENDIX A

3/3

**SHORT**      **RetrieveBin**(STRING binID, STRING time)  
 Given a location (binID), the carousel will rotate the carriers, if necessary, and open the bin doors for retrieval. The doors will close in "time" seconds.

**Parameter Format:**

binID = U000000B00000

time = {0 ... 65} seconds, 0 = forever, default = 30 seconds

**Return Format:**

0 - Successfully sent information to Carousel Driver

-1 - Socket error (call GetErrDesc() to get description of error)

**SHORT**      **Rotate** (STRING carrier, STRING shelf)  
 The carousel will rotate to the specified location determined by carrier and shelf.

**Parameter Format:**

Position = {1 ... Max shelves}

**Return Format:**

0 - Successfully sent information to Carousel Driver

-1 - Socket error (call GetErrDesc() to get description of error)

**SHORT**      **RotateDown**()  
 The carousel will rotate down one carrier.

**Return Format:**

0 - Successfully sent information to Carousel Driver

-1 - Socket error (call GetErrDesc() to get description of error)

**SHORT**      **RotateUp**()  
 The carousel will rotate up one carrier.

**Return Format:**

0 - Successfully sent information to Carousel Driver

-1 - Socket error (call GetErrDesc() to get description of error)



**Customer Interface API****Overview**

The Customer Interface is the set of functions used by a NC application to provide a carousel service to a customer. The Customer Interface has access to the Carousel Driver through an immediate response port (command respond port) or a process and generate event port (command process port). The command respond port will return with the function result. The command process port will return the success of sending the message and upon receiving the completion or error of a command, this port will generate an event with the status of the last command.

Events will be returned as one of the format: MsgEvent(INT msgVal, STRING tag)

Current values are as follows:

- -2 - Socket Error (use GetErrDesc() for description)
- -1 - Socket Error (not connected to Carousel Driver)
- 0 - OK: Operation complete
- 1 - HWERR: Hardware error that reports carousel errors
- 2 - ERR: error encountered while executing command (use GetErrDesc() for description)
- 3 - TIMEOUT: Result not received by the Driver within time period
- 4 - INUSE: Driver in use by the employee
- 5 - BUSY: Carousel Driver is busy processing another command

Registry values (in HKEY\_CLASSES\_ROOT\CUPSS):

- |                       |   |
|-----------------------|---|
| CMD_PROCESS_TIMEOUT   | - Time period to wait for result from driver before returning "timed out" error.<br>- If "0", then timeout is not used. |
| CMD_RESPOND_TIMEOUT   | - Time period to wait for result from driver before returning "timed out" error.<br>- If "0", then timeout is not used. |
| CUST_CMD_PROCESS_PORT | - Port number of the command process port (i.e. 5007)   |
| CUST_CMD_RESPOND_PORT | - Port number of the command respond port (i.e. 5009)   |
| CUST_TAKE_CONTROL     | - If "1", then customer will take control of carousel from employee.  |
| DRIVER_HOST_NAME      | - Host name where driver is running (i.e. "127. 0. 0. 0" or "localhost")  |

**Carousel Functions:**

**SHORT** CloseDoors()

The carousel will close the doors that are currently open.

Return Format:

- 0 - Successfully sent information to Carousel Driver
- 1 - Socket error (call GetErrDesc() to get description of error)

**STRING** GetBinInfo(STRING binID)

Returns the carrier number, height, width, depth, and unit number of the specified location (binID).

Parameter Format:

BinID = U000000B00000

Return Format:

Unit:xx, Carrier:yy, Shelf:ss, H:hh, W:ww, D:dd

Where xx = Unit Number  
yy = Carrier Number  
ss = Shelf Number  
hh = height (inches)  
ww = width (inches)  
dd = depth (inches)

**STRING** GetCtrlVersion()

Returns the current version of the ActiveX control.

Return Format:

x.x (i.e. "0.9")

## APPENDIX B

2/3

**STRING**      **GetDrvVersion()**  
 Returns the current version of the carousel driver.  
 Return Format:  
     x.x (i.e. "0.9")

**STRING**      **GetErrDesc()**  
 Returns an error description of the last function.  
 Return Format:  
     Error description (i.e. "Carousel Busy", "In Use by Customer")

**SHORT**      **GetMaxBins (STRING carrier, STRING shelf)**  
 If "carrier" and "shelf" are empty, then this method returns the maximum number of bins in current carousel configuration. Otherwise, the number of bins for the specified carrier and shelf are returned.  
 Return Format:  
     Max bins in carousel {i.e. 114}  
     OR  
     Max bins per carrier and shelf {i.e. 7}

**SHORT**      **GetMaxCarriers ()**  
 Returns the maximum number of carriers in current carousel configuration.  
 Return Format:  
     Max carriers {i.e. 12}

**SHORT**      **GetMaxShelves (STRING carrier)**  
 If "carrier" is empty, then this method returns the maximum number of shelves in current carousel configuration. Otherwise, the number of shelves on the specified carrier is returned.  
 Return Format:  
     Max shelves {i.e. 36}  
     OR  
     Max shelves per carrier {i.e. 3}

**STRING**      **GetState()**  
 Returns the current state of the customer access to the carousel.  
 Return Format:  
     {busy, error, idle, inuse, ready}

**SHORT**      **Initialize()**  
 Initialize the socket connections using registry values EMP\_CMD\_PROCESS\_PORT, EMP\_CMD\_RESPOND\_PORT and DRIVER\_HOST\_NAME.  
 Return Format:  
     Bit map of errors  
     1 - process port invalid  
     2 - respond port invalid  
     3 - no host name for driver

## APPENDIX B

3/3

SHORT QueueBin(String BinID)

The carousel will rotate the carriers to put the location (BinID) in a retrieval position.

Parameter Format:

BinID = U000000B00000

Return Format:

0 - Successfully sent information to Carousel Driver  
-1 - Socket error (call GetErrDesc() to get description of error)

STRING RequestService()

Request that employee functions be terminated and customer be given control of the carousel if registry value CUST\_TAKE\_CONTROL is set to "1". Currently not implemented.

Return Format:

0 - Successfully sent information to Carousel Driver  
-1 - Socket error (call GetErrDesc() to get description of error)

SHORT RetrieveBin(String binID, String time)

Given a location (binID), the carousel will rotate the carriers, if necessary, and open the bin doors for retrieval. The doors will close in "time" seconds.

Parameter Format:

binID = U000000B00000  
time = { 1 ... 65 } seconds, default = 30 seconds

Return Format:

0 - Successfully sent information to Carousel Driver  
-1 - Socket error (call GetErrDesc() to get description of error)

## APPENDIX C

1215-0422P  
BD-01-029

1/4

**Carousel Driver Interfaces****Overview**

The Carousel Driver is an executable program that communicates directly with the carousel. Both customer and employee interface controls will communicate with the carousel via this driver. The "tag" is optional for process commands, but recommended for event processing by the ActiveX controls (Customer and Employee Interface).

Possible result message returned to controls: ...

- Operation complete
- In use by Customer
- In use by Employee
- Function not supported
- Cannot access database (specified in registry value CAROUSEL\_DB)
- Bin ID incorrect format ("U000000B00000")
- Bin ID not found
- Bin Type not found
- Timed out while waiting for carousel result
- Carousel Out of Order
- Carousel Obstruction
- Obstruction cleared
- Communications error with carousel

Registry values:

CAROUSEL_COM_PORT	- Serial communications port number (i.e. "1" for COM1).
CAROUSEL_COM_SETTINGS	- Serial communications port settings in the format "baud rate, parity, bits, start bit" (i.e. "9600, n, 8, 1").
CAROUSEL_DB	- Complete path and name of the carousel configuration database.
CAROUSEL_TIMEOUT	- Period to wait for carousel result before returning "timed out" error. - If "0", then timeout is not used.
INACTIVITY_TIMEOUT	- Close socket communications with controls after specified time of inactivity. - If "0", then timeout is not used.

**Customer Interface:****Close Doors**

Send: "close[:tag] <cr>"  
where tag = <optional>": ..."

**Get Bin Information**

Send: "getbininfo,[BinID] [:tag] <cr>"  
where BinID = U000000B00000  
tag = <optional>": ..."

Response: "U:xx,C:yy,S:ss,H:hh,W:ww,D:dd[:tag]<cr>"  
where xx = Unit Number  
yy = Carrier Number  
ss = Shelf Number  
hh = height (inches)  
ww = width (inches)  
dd = depth (inches)  
tag = ": ..." (If provided)

**Get Current State**

Send: "getstate[:tag]<cr>"  
where tag = <optional>": ..."

Response: "xx[:tag]<cr>"  
where xx = {idle, busy, inuse}  
tag = ": ..." (If provided)

## APPENDIX C

2/4

## Get Error Description

Send: "geterrordesc[:tag]<cr>"  
       where tag = <optional>": ..."

Response: "xx[:tag]<cr>"  
       where xx = Error Description Text  
       tag = ": ..." (If provided)

## Get Max Bins

Send: "getmaxbins,[carrier],[shelf][:tag]<cr>"  
       where carrier = <optional> {1 ... Maximum number of carriers}  
       shelf = <optional> {1 ... Maximum number of shelves per carrier}  
       tag = <optional>": ..."

Response: "xx[:tag]<cr>"  
       where xx = Maximum number of bins per shelf (If [carrier] and [shelf] are specified.)  
       tag = ": ..." (If provided)  
       OR  
       where xx = Maximum number of bins in carousel  
       tag = ": ..." (If provided)

## Get Max Carrier

Send: "getmaxcarriers[:tag]<cr>"  
       where tag = <optional>": ..."

Response: "xx[:tag]<cr>"  
       where xx = Maximum number of carriers  
       tag = ": ..." (If provided)

## Get Max Shelves

Send: "getmaxshelves,[carrier][:tag]<cr>"  
       where carrier = <optional> {1 ... Maximum number of carriers}  
       tag = <optional>": ..."

Response: "xx[:tag]<cr>"  
       where xx = Maximum number of shelves per carrier (If [carrier] is specified.)  
       tag = ": ..." (If provided)  
       OR  
       where xx = Maximum number of shelves in carousel  
       tag = ": ..." (If provided)

## Queue a Bin Location

Send: "que,[binID][:tag]<cr>"  
       where binID = U000000B00000  
       tag = <optional>": ..."

## Open a Bin Location

Send: "retrieve,[binID],[time][:tag] <cr>"  
       where binID = U000000B00000  
       time = <optional>1-65 seconds, default = 30 seconds  
       tag = <optional>": ..."

## Request Service

Send: "request[:tag] <cr>"  
       where tag = <optional>": ..."

## APPENDIX C

3/4

**Employee Interface:****Close Doors**

Send: "close[:tag] <cr>"  
 where tag = <optional>": ..."

**Get Bin Information**

Send: "getbininfo,[BinID] [:tag] <cr>"  
 where BinID = U000000B00000  
 tag = <optional>": ..."

Response: "U:xx, C:yy, S:ss, H:hh, W:ww, D:dd[:tag] <cr>"  
 where xx = Unit Number  
 yy = Carrier Number  
 ss = Shelf Number  
 hh = height (inches)  
 ww = width (inches)  
 dd = depth (inches)  
 tag = ": ..." (If provided)

**Get Current State**

Send: "getstate[:tag] <cr>"  
 where tag = <optional>": ..."

Response: "xx[:tag] <cr>"  
 where xx = {idle, busy, inuse}  
 tag = ": ..." (If provided)

**Get Error Description**

Send: "geterrordesc[:tag] <cr>"  
 where tag = <optional>": ..."

Response: "xx[:tag] <cr>"  
 where xx = Error Description Text  
 tag = ": ..." (If provided)

**Get Max Bins**

Send: "getmaxbins,[carrier],[shelf][:tag] <cr>"  
 where carrier = <optional> {1 ... Maximum number of carriers}  
 shelf = <optional> {1 ... Maximum number of shelves per carrier}  
 tag = <optional>": ..."

Response: "xx[:tag] <cr>"  
 where xx = Maximum number of bins per shelf (If [carrier] and [shelf] are specified.)  
 tag = ": ..." (If provided)

OR

where xx = Maximum number of bins in carousel  
 tag = ": ..." (If provided)

**Get Max Carrier**

Send: "getmaxcarriers[:tag] <cr>"  
 where tag = <optional>": ..."

Response: "xx[:tag] <cr>"  
 where xx = Maximum number of carriers  
 tag = ": ..." (If provided)

## APPENDIX C

4/4

## Get Max Shelves

Send: "getmaxshelves,[carrier][:tag] <cr>"  
 where carrier = <optional> {1 ... Maximum number of carriers}  
 tag = <optional> ": ..."

Response: "xx[:tag] <cr>"  
 where xx = Maximum number of shelves per carrier (If [carrier] is specified.)  
 tag = ": ..." (If provided)

OR  
 where xx = Maximum number of shelves in carousel  
 tag = ": ..." (If provided)

## Open a Bin Location

Send: "retrieve,[binID],[time][:tag] <cr>"  
 where binID = U000000B00000  
 time = <optional> 0-65 seconds, 0 => forever, default = 30 seconds  
 tag = <optional> ": ..."

## Open All Doors

Send: "openall[:tag] <cr>"  
 where tag = <optional> ": ..."

## Rotate Down one Carrier

Send: "rotatedown[:tag] <cr>"  
 where tag = <optional> ": ..."

## Rotate Up one Carrier

Send: "rotateup[:tag] <cr>"  
 where tag = <optional> ": ..."

## Rotate to Shelf

Send: "rotate,[carrier],[shelf][:tag] <cr>"  
 where carrier = {1 ... Maximum number of carriers}  
 shelf = {1 ... Maximum number of shelves per carrier}  
 tag = <optional> ": ..."

What is claimed is:

1. A web enabled item delivery and retrieval system, comprising:

a storage subsystem including a secure storage facility accessible via software control employing browser technology by a first user who loads and stores an item into a storage location with a first identifier as to the storage location and a second identifier as to the identity of a second user, said second user then retrieving said item or returning an item upon using and entering certain information into an access terminal located on the storage facility; and

a computer subsystem which controls the storage facility and having an application configurable software control architecture including a browser software interface including object-oriented programs comprising, a storage facility driver software interface for controlling access to the storage facility, a back office application program interface (BO API) enabling the first user to access the storage facility by means of the driver software interface, and a front office application program interface (FO API) enabling the second user to access the storage facility also by means of the driver software interface; and

wherein said secure storage facility includes comprises a carousel and controls therefore, and including a plurality of storage bins normally hidden behind a closed door assembly including a plurality of doors, said doors being selectively opened on demand by either the first user via the back office application program interface (BO API) or the second user via the front office application program interface (FO API).

2. The system according to claim 1 wherein the carousel comprises a vertical carousel.

3. The system according to claim 1 wherein the first and second user commonly use said access terminal, said access terminal having a screen supporting a web page display.

4. The system according to claim 3 wherein the system comprises a single sided system where the carousel is accessed from a front side by both the first and second user.

5. The system according to claim 1 wherein the first and second user use separate access terminals, said terminals each having a screen supporting a web page display.

6. The system according to claim 5 wherein the system comprises a double sided system where the carousel is accessed from a rear side by the first user and from a front side by the second user.

7. The system according to claim 1 wherein the carousel provides access from a front side and wherein the door assembly includes a set of doors including at least one door on the front side of the carousel which is accessible only by the first user and at least one door on the front side of the carousel which is accessible only by the second user and wherein the common user access terminal is located on the front side of the carousel.

8. The system according to claim 7 wherein the carousel also provides access from a back side and wherein the door assembly includes at least one door on the back side which is accessible only by the first user.

9. The system according to claim 1 wherein the first user comprises an employee of a service company and the second user comprises a customer of the service company.

10. The system according to claim 1 wherein the first user comprises respective employees of a plurality of delivery service companies, said delivery service companies inserting respective application software into the computer subsystem which is executed in turn so to provide exclusive use

of the storage facility at any one time by said plurality of delivery service companies.

11. The system according to claim 1 wherein said access terminal is located on a kiosk.

12. The system according to claim 11 wherein the kiosk houses the browser software interface.

13. The system according to claim 12 wherein the kiosk is located at the front of the carousel.

14. The system according to claim 13 wherein the kiosk supports a touch screen for inputting user access information.

15. The system according to claim 13 wherein the kiosk supports a signature pad for inputting a user signature.

16. The system according to claim 13 wherein the kiosk supports a bar code reader for inputting user bar code information.

17. The system according to claim 13 wherein the kiosk supports a card reader for inputting user card information.

18. The system according to claim 13 wherein the kiosk supports a PIN pad for inputting a user PIN number.

19. The system according to claim 13 wherein said kiosk supports a receipt printer for printing a user transaction receipt.

20. The system according to claim 1 and additionally including a wireless communications device for accessing the storage subsystem and the computer subsystem via a local area network.

21. The system according to claim 1 and additionally including a handheld wireless communications device for accessing the storage subsystem and the computer subsystem.

22. The system according to claim 1 and additionally including a wireless handheld communications device having a screen and incorporating a scanner for accessing the storage subsystem and the computer subsystem.

23. The system according to claim 1 wherein said software architecture additionally includes a security software interface for controlling a camera system for taking a picture of a user while interacting with the browser interface while at the storage subsystem.

24. The system according to claim 23 wherein the user comprises the second user.

25. The system according to claim 23 wherein the security software interface includes application interface controls.

26. The system according to claim 1 and additionally including an application and data web page server connectable to the browser interface.

27. The system according to claim 1 and additionally including an application and data web page server connectable to the browser software interface and a master web page server connectable to the application and data web page server which supports and stores one or more sets of web pages for said web page display.

28. The system according to claim 1 wherein the object oriented programs of the back office application program interface (BO API) implement functions in a basic carousel control architecture during an item loading operation, comprising:

a connect function which initializes connections of the object oriented programs of the back office application program interface to the driver software interface and passes an identification code thereto, if necessary, for access control;

an open all doors function gains full access to the carousel;

an open bin location function to position the carousel and open the doors to a specific bin;



39

a rotate carousel function which positions the carousel to a predetermined bin access point for a loading operation;

an identify bin function which is used to identify a particular bin when all the doors are open;

a close bin function which is used to close all the doors and, if necessary, clear all bin access codes; and

a close all doors function which closes all doors and secures the carousel so as to complete an item loading transaction.

29. The system according to claim 1 wherein the object oriented program of the front office application program interface (FO API) implemented functions in a basic carousel control architecture during an item retrieval operation, comprising:

a connect function which initializes connections of the object oriented programs of the front office application program interface to the driver software interface and passes an identification code thereto, if necessary, for access control security;

a cue bin location function which rotates the carousel such that one requested bin is positioned behind a door of said door assembly without any of the doors being opened while an authentication process takes place;

an open bin location function to open said door to the requested bin for item retrieval; and

a close bin function which is thereafter used to close said door so as to complete an item retrieval transaction.

30. A system according to claim 1 wherein the object oriented programs of the back office application program interface (BO API) implements functions during an item loading operation, comprising:

a connect function which initializes connection of the object oriented programs of the back office application program interface to the driver software interface;

a register item function which registers a specific item to be loaded in the carousel in an inventory database;

a purge item function which removes an item in the inventory database and clears a bin access code therefor;

a load item function which positions the carousel and opens a door of the carousel for a specific item at a specific location;

a removal item function which positions the carousel and opens the door to a specific item for removal and which is then marked as removed from the inventory database;

an open all doors function which is used to gain full access to the carousel;

an open bin location function similar to the load item function and positions the carousel to a specified bin and opens the doors thereto;

an identify bin function which identifies a particular bin when all the doors of the carousel are opened;

a rotate carousel function which is used to position the carousel to a specific access point;

a close bin function which is used to close the door for a specific bin location;

a close all doors function which is used to close all doors and secure the machine; and

a database maintenance and report function to update the inventory database.

31. The system according to claim 1 wherein the front office application program interface (FO API) implements functions during a retrieval or return operation, comprising:

40

a connect function which initializes connections of the object oriented programs of the front office application program interface to the driver software interface;

a cue item and authenticate user function which rotates the carousel such that a requested item for retrieval is positioned behind a specific door without any of the doors being opened while a transactional process of authenticating the user takes place;

a remove item function which positions the carousel and opens a door to a specified item for retrieval;

a close bin function which is used to close doors of the carousel;

a load item function which positions the carousel and opens a door for return of a specified item at a specific bin location where the item is then registered in an inventory database;

a return item function which closes the door of the carousel upon return of an item to a specified bin and which is flagged in the inventory database for return;

a query item function to find and load time and status information into the inventory database; and

a print receipt function to print a receipt of a transaction carried out by a user.

32. A web enabled item delivery and retrieval system, comprising:

a storage subsystem including a secure storage facility accessible by a first user who loads and stores an item into a storage location with a first identifier as to the storage location and a second identifier as to the identity of a second user, said second user then retrieving said item or returning an item upon using and entering certain information into an access terminal; and

a computer subsystem which controls the storage facility and having an application configurable software control architecture including a software interface including object-oriented programs comprising, a storage facility driver software interface for controlling access to the storage facility, a back office application program interface (BO API) enabling the first user to access the storage facility by means of the driver software interface, and a front office application program interface (FO API) enabling the second user to access the storage facility also by means of the driver software interface; and

wherein said secure storage facility includes comprises a carousel and controls therefore, and including a plurality of storage bins normally hidden behind a closed door assembly including a plurality of doors, said doors being selectively opened on demand by either the first user via the back office application program interface (BO API) or the second user via the front office application program interface (FO API);

wherein the back office application program interface (BO API) implements functions during an item loading operation, comprising:

a connect function which initializes connection of the application controls of the back office application program interface to the carousel driver;

a register item function which registers a specific item to be loaded in the carousel in an inventory database;

a purge item function which removes an item in the inventory database and clears a bin access code therefor;

a load item function which positions the carousel and opens a door of the carousel for a specific item at a specific location;

41

- a removal item function which positions the carousel and opens the door to a specific item for removal and which is then marked as removed from the inventory database;
  - an open all doors function which is used to gain full access to the carousel; 5
  - an open bin location function similar to the load item function and positions the carousel to a specified bin and opens the doors thereto;
  - an identify bin function which identifies a particular bin when all the doors of the carousel are opened; 10
  - a rotate carousel function which is used to position the carousel to a specific access point;
  - a close bin function which is used to close the door for a specific bin location; 15
  - a close all doors function which is used to close all doors and secure the machine; and a database maintenance and report function to update the inventory database.
33. A web enabled item delivery and retrieval system according to claim 32 wherein the access terminal is located in a kiosk containing the software interface and wherein the kiosk is secured to the storage facility. 20
34. A web enabled item delivery and retrieval system, comprising: 25
- a storage subsystem including a secure storage facility accessible by a first user who loads and stores an item into a storage location with a first identifier as to the storage location and a second identifier as to the identity of a second user, said second user then retrieving said item or returning an item upon using and entering certain information into an access terminal; and 30
  - a computer subsystem which controls the storage facility and having an application configurable software control architecture including a software interface including object-oriented programs comprising, a storage facility driver software interface for controlling access to the storage facility, a back office application program interface (BO API) enabling the first user to access the storage facility by means of the driver software interface, and a front office application program interface (FO API) enabling the second user to access the 35

42

- storage facility also by means of the driver software interface; and
  - wherein said secure storage facility includes comprises a carousel and controls therefore, and including a plurality of storage bins normally hidden behind a closed door assembly including a plurality of doors, said doors being selectively opened on demand by either the first user via the back office application program interface (BO API) or the second user via the front office application program interface (FO API);
  - wherein the front office application program interface (FO API) implements functions during a retrieval or return operation, comprising:
    - a connect function which initializes connections of the application controls of the front office application program interface to the carousel driver interface;
    - a cue item and authenticate user function which rotates the carousel such that a requested item for retrieval is positioned behind a specific door without any of the doors being opened while a transactional process of authenticating the user takes place;
    - a remove item function which positions the carousel and opens a door to a specified item for retrieval;
    - a close bin function which is used to close doors of the carousel;
    - a load item function which positions the carousel and opens a door for return of a specified item at a specific bin location where the item is then registered in an inventory database;
    - a return item function which closes the door of the carousel upon return of an item to a specified bin and which is flagged/marked in the inventory database for return;
    - a query item function to find and load time and status information into the inventory database; and
    - a print receipt function to print a receipt of a transaction carried out by a user.
35. A web enabled item delivery and retrieval system according to claim 34 wherein the access terminal is located on a kiosk containing the software interface and wherein the kiosk is secured to the storage facility. 40

\* \* \* \* \*



US006424841B1

(12) **United States Patent**  
**Gustafsson**

(10) **Patent No.:** **US 6,424,841 B1**  
(45) **Date of Patent:** **Jul. 23, 2002**

(54) **SHORT MESSAGE SERVICE WITH  
IMPROVED UTILIZATION OF AVAILABLE  
BANDWIDTH**

(75) **Inventor:** **Patrik G. Gustafsson**, Foster City, CA  
(US)

(73) **Assignee:** **Openwave Systems Inc.**, Redwood  
City, CA (US)

(\*) **Notice:** Subject to any disclaimer, the term of this  
patent is extended or adjusted under 35  
U.S.C. 154(b) by 0 days.

(21) **Appl. No.:** **09/252,960**

(22) **Filed:** **Feb. 18, 1999**

(51) **Int. Cl.<sup>7</sup>** ..... **H04B 1/04; H04Q 7/20**

(52) **U.S. Cl.** ..... **455/466; 455/466; 455/417**

(58) **Field of Search** ..... **455/422, 450,**  
**455/466, 517, 426; 370/428, 338, 394;**  
**709/203, 227; 707/10**

(56) **References Cited**

#### U.S. PATENT DOCUMENTS

5,481,539 A	1/1996	Hershey et al.	
5,579,372 A	11/1996	Åström	
5,692,032 A	11/1997	Seppänen et al.	
5,748,100 A *	5/1998	Gutman et al.	340/825.44
5,751,708 A	5/1998	Eng et al.	
5,809,415 A	9/1998	Rossmann	
5,930,239 A *	7/1999	Turcotte	370/310

6,108,325 A *	8/2000	Stephanson et al.	370/337
6,119,167 A	9/2000	Boyle et al.	
6,138,158 A	10/2000	Boyle et al.	
6,163,546 A *	12/2000	Sipila	370/446
6,185,208 B1 *	2/2001	Liao	370/392

#### OTHER PUBLICATIONS

Global System for Mobile Communications (GSM) 03.40,  
version 5.6.1, European Telecommunications Standards  
Institute (ETS) (ETS 300 901), Jan. 1998.

\* cited by examiner

*Primary Examiner*—Edward F. Urban

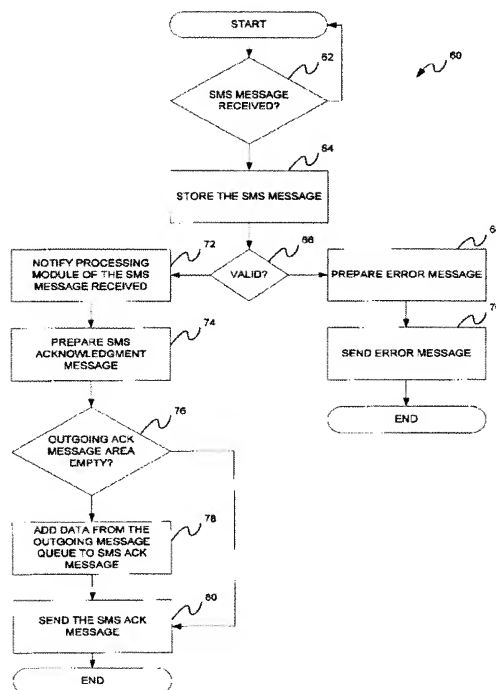
*Assistant Examiner*—Meless Zewdu

(74) *Attorney, Agent, or Firm*—Beyer Weaver & Thomas,  
LLP

(57) **ABSTRACT**

Techniques that enable wireless client devices to more  
efficiently utilize available transmission bandwidth in a  
wireless network are disclosed. In one embodiment, the  
techniques operates to include or incorporate return infor-  
mation (data) in an acknowledgement message after an  
incoming message has been successfully received from a  
sender. As a result, messages are able to be effectively sent  
over the wireless network with less overhead. The tech-  
niques are particularly suitable for a Global System for  
Mobile Communications (GSM) network that is capable of  
bi-directional communications with a short message service  
center (SMSC).

**28 Claims, 14 Drawing Sheets**



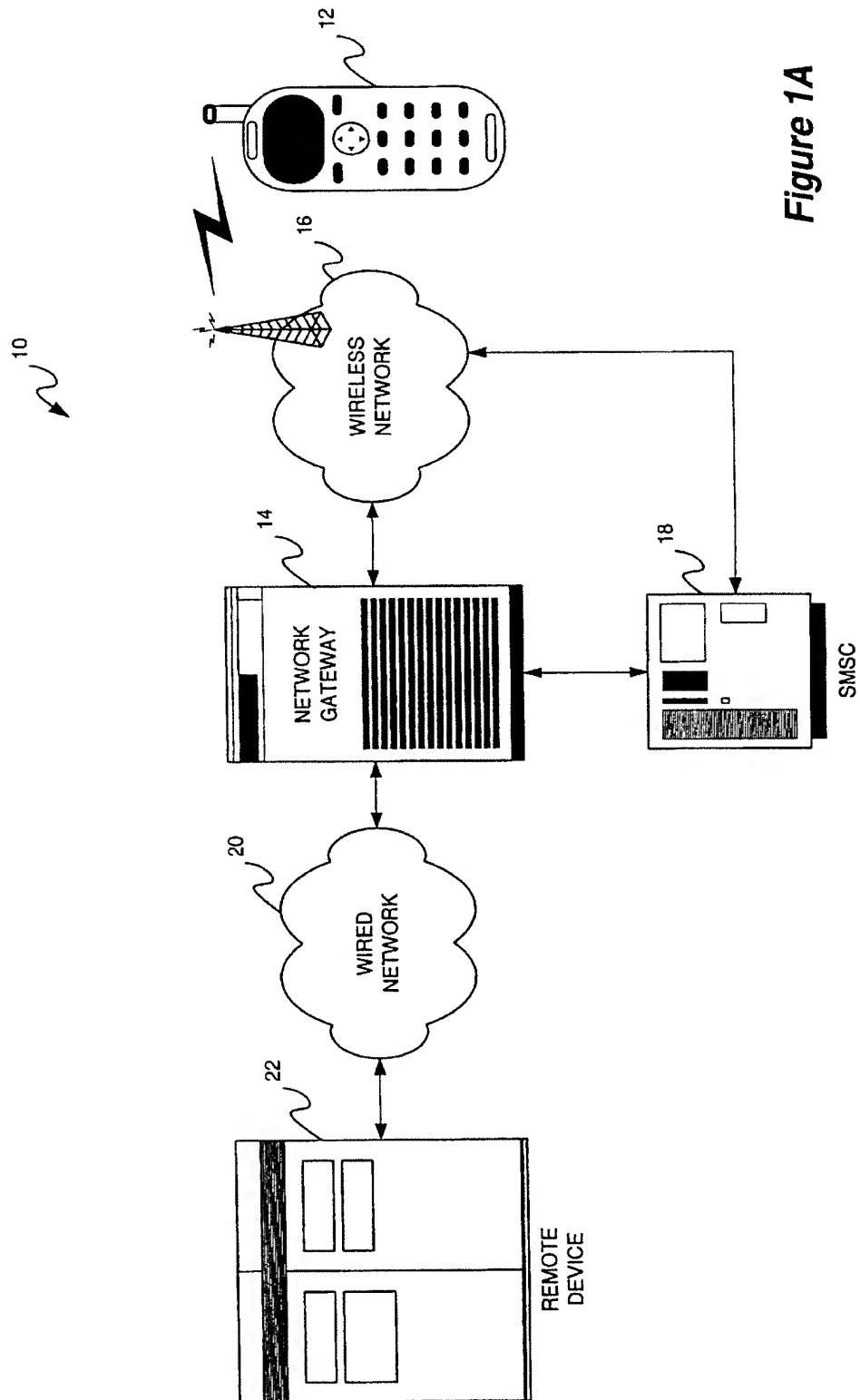


Figure 1A

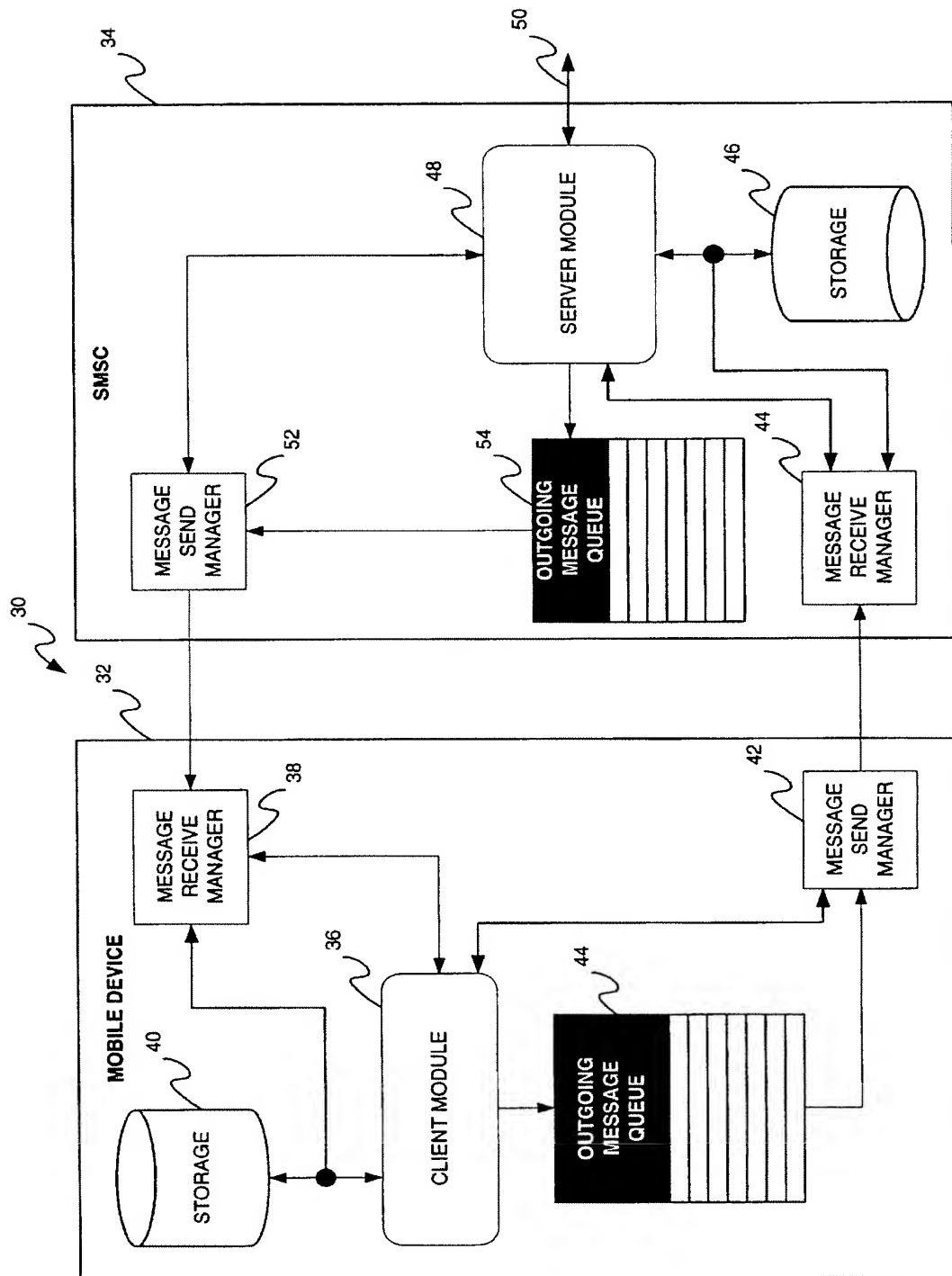
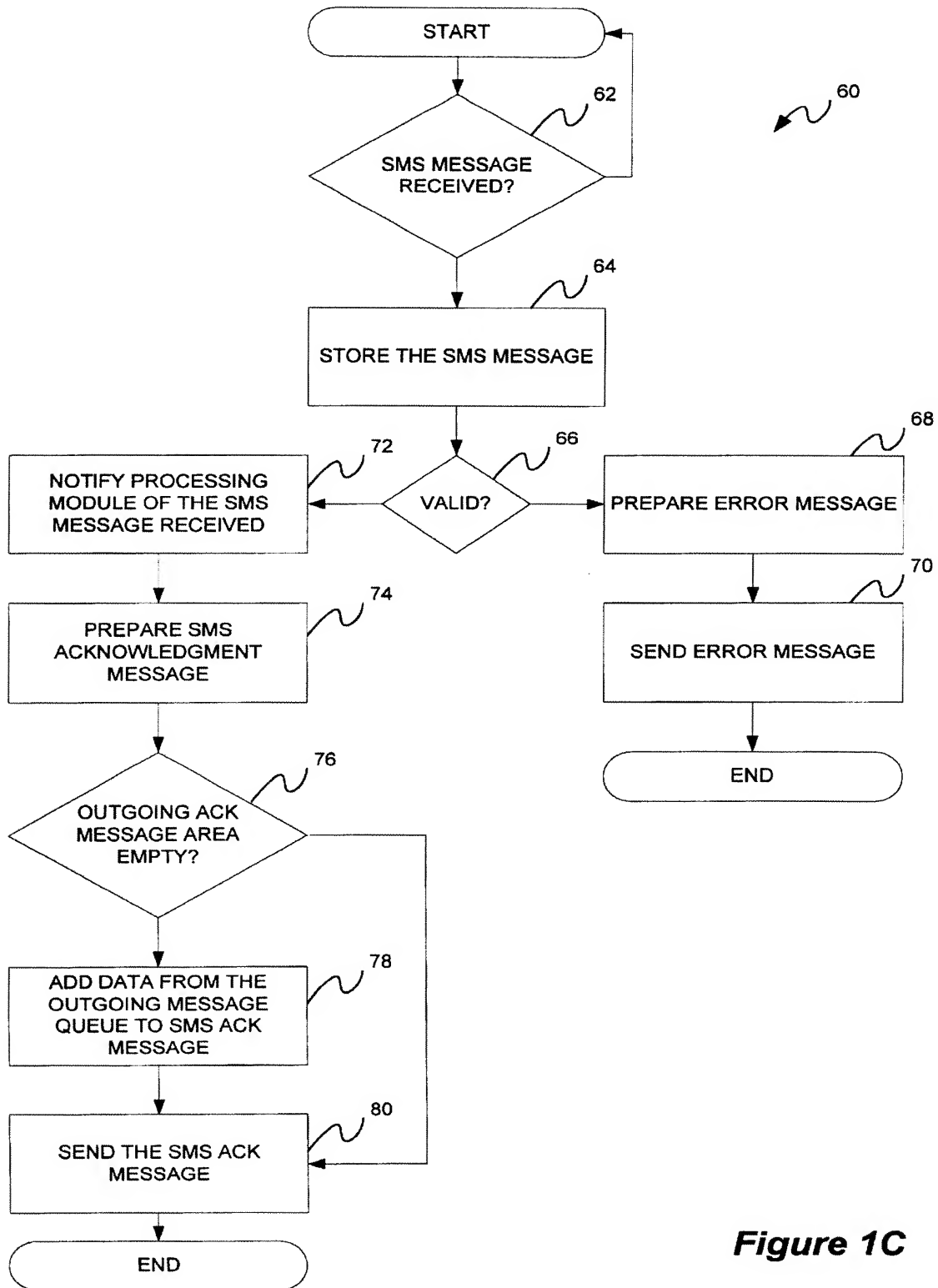


Figure 1B

**Figure 1C**

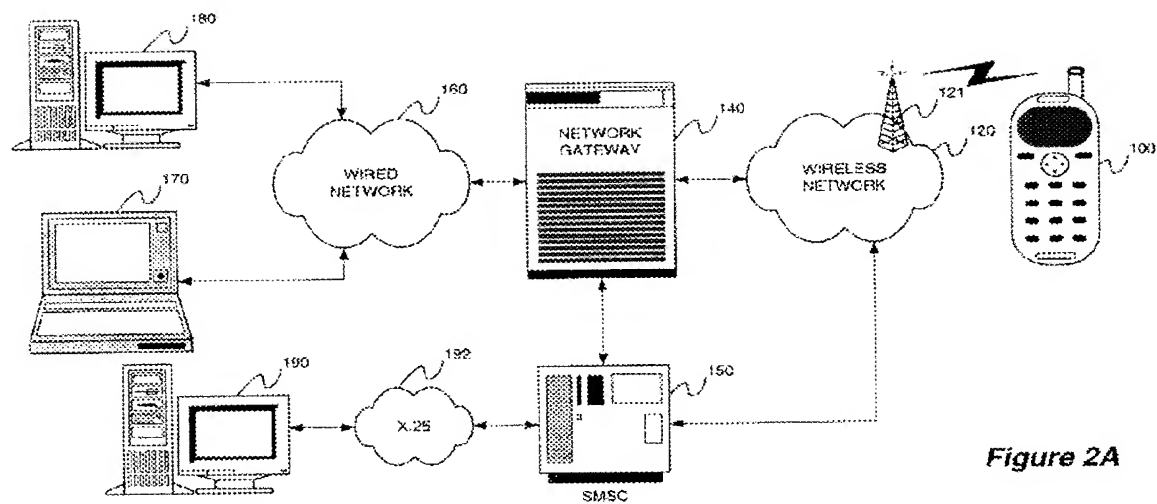
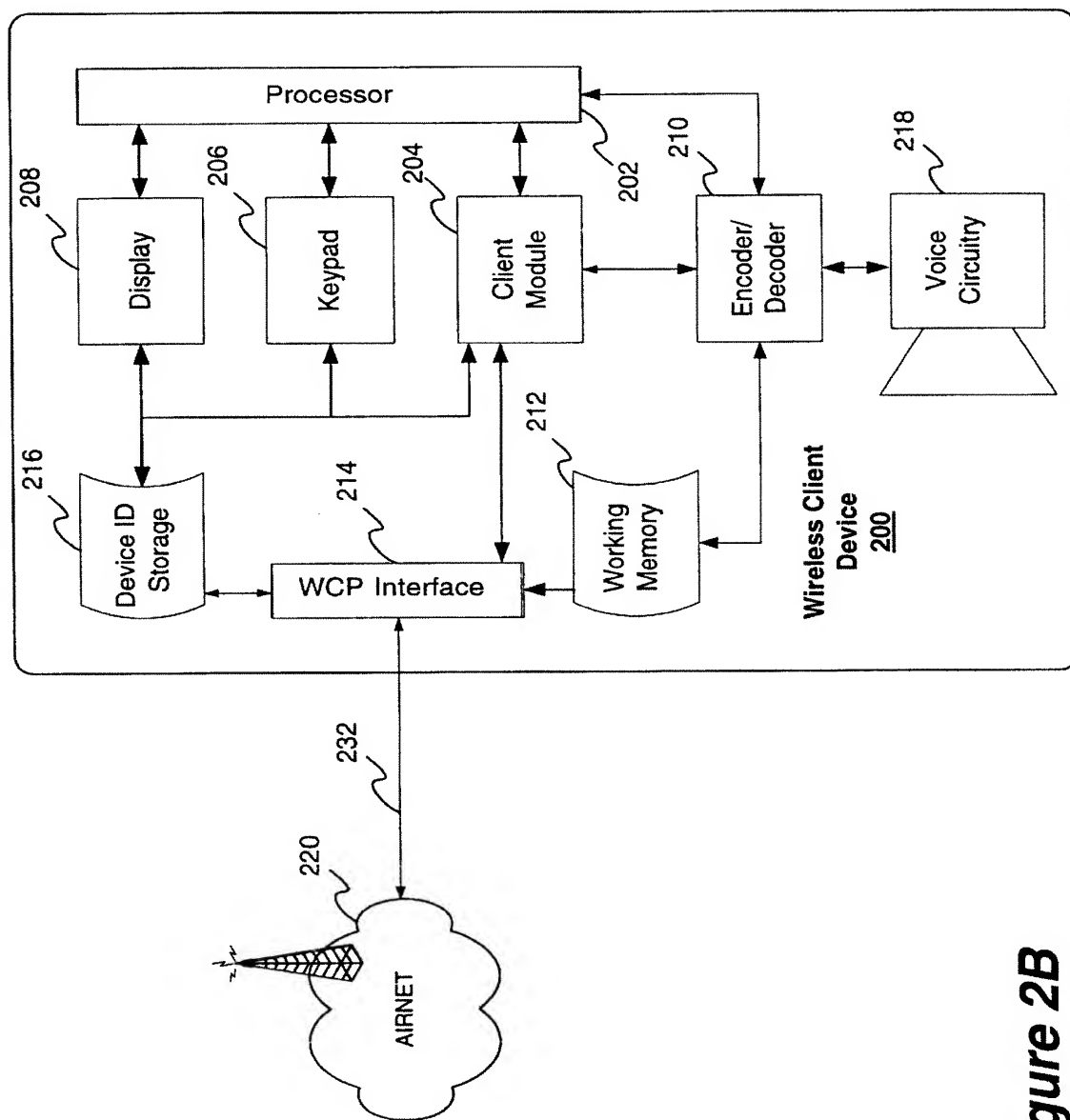


Figure 2A

**Figure 2B**



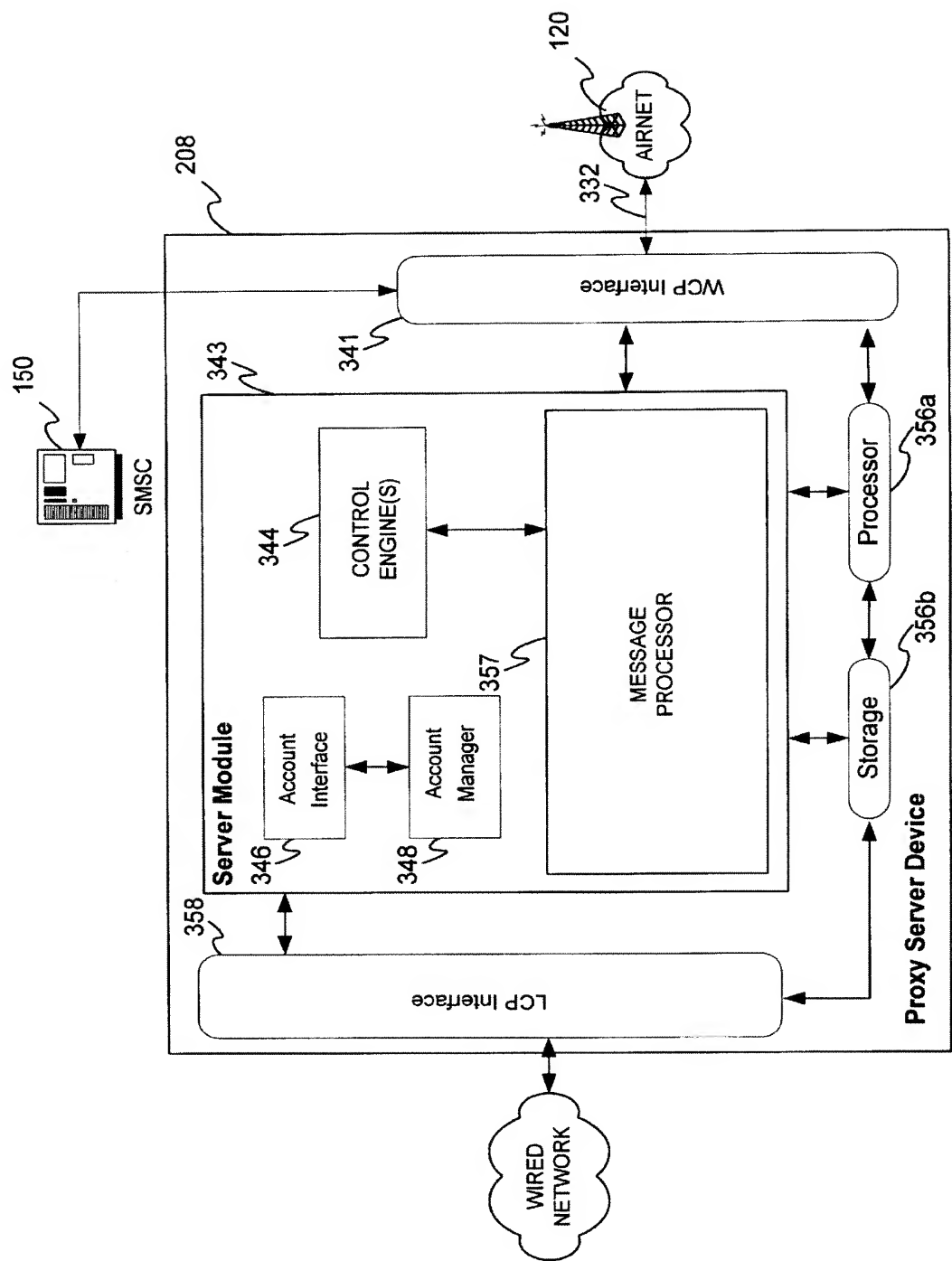


Figure 3

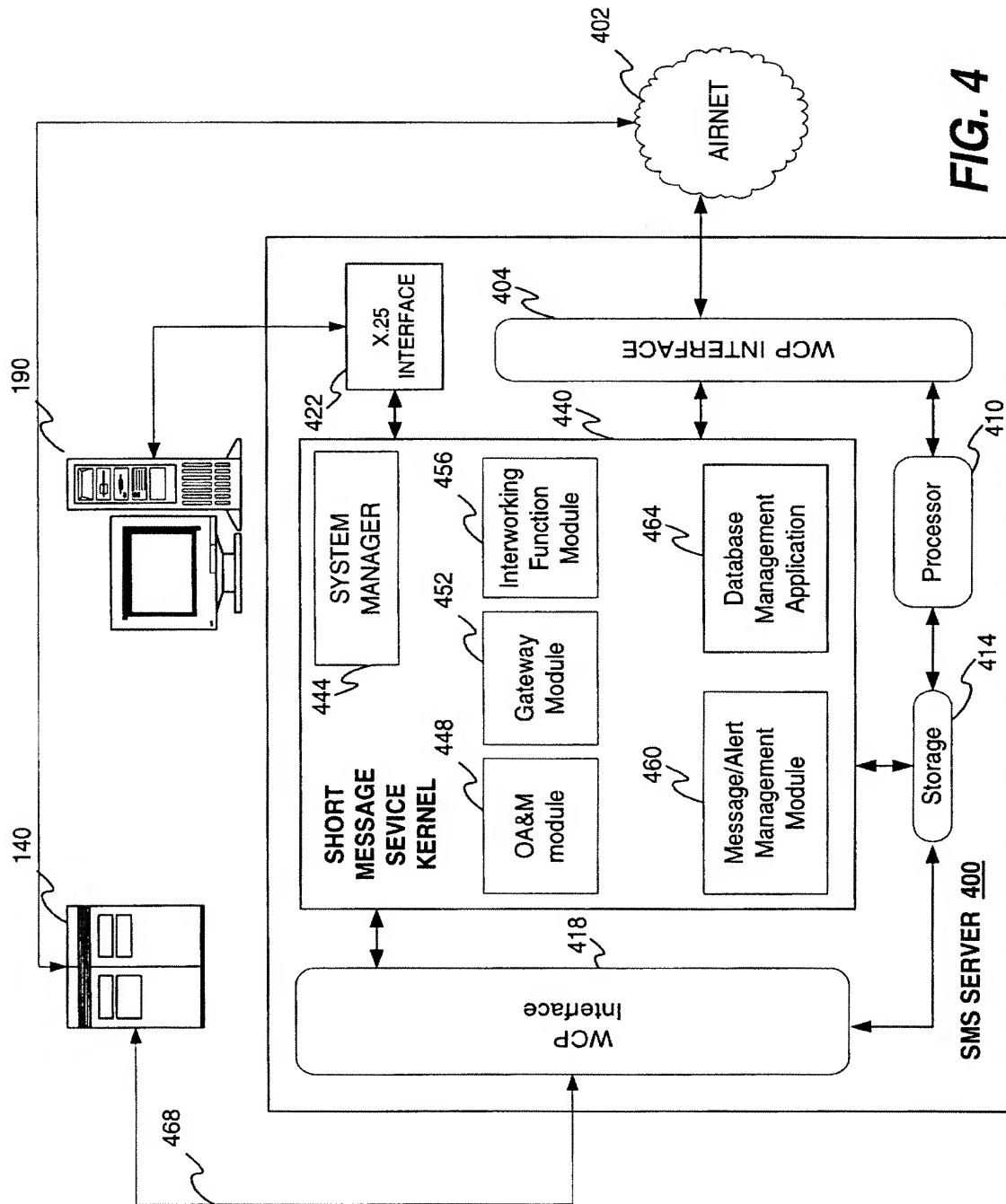
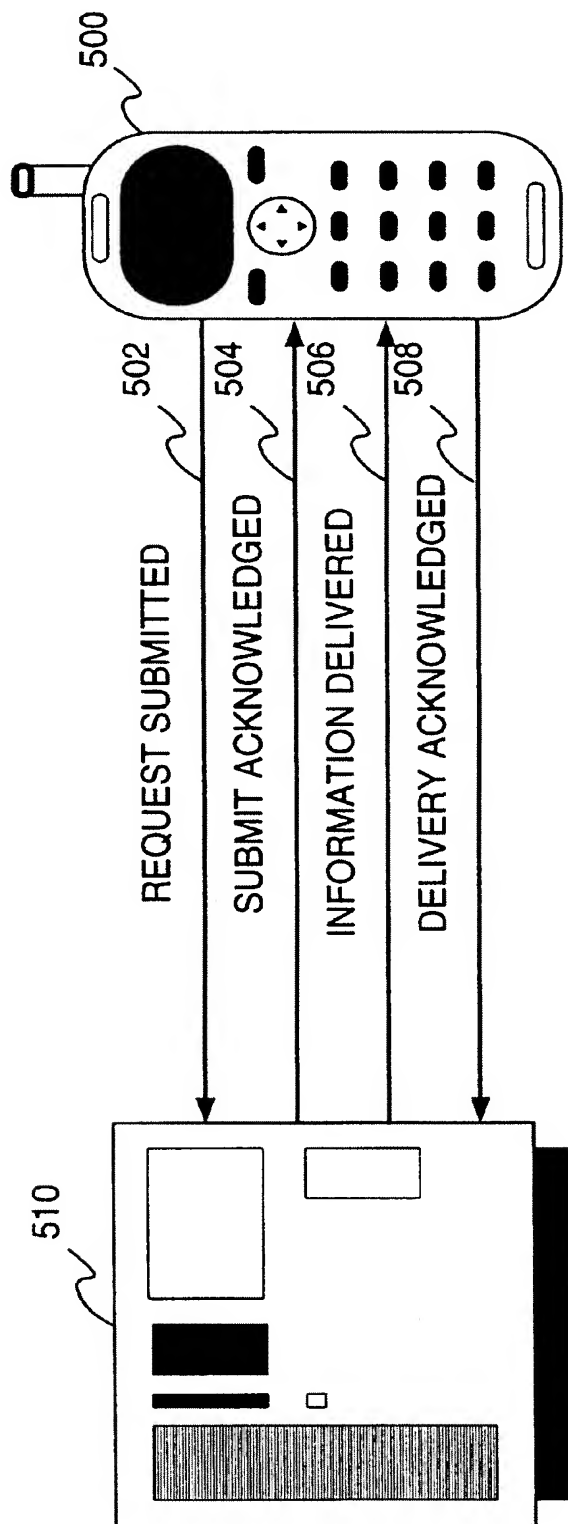
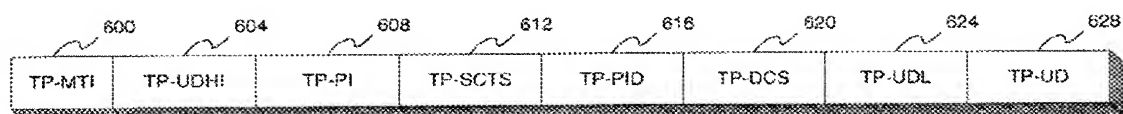


FIG. 4



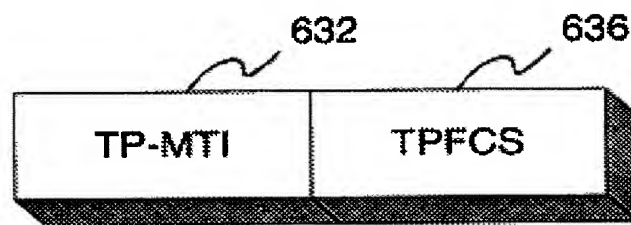
**FIG. 5**

SUBMIT SUCCESS REPORT



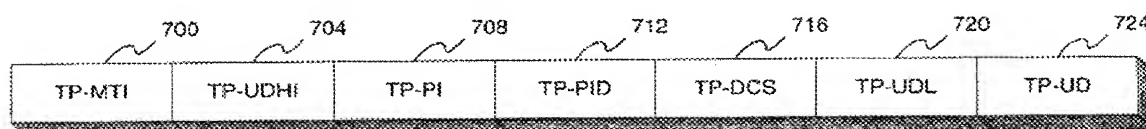
**FIG. 6A**

SUBMIT ERROR REPORT

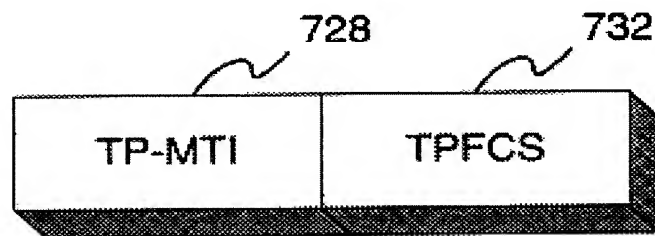


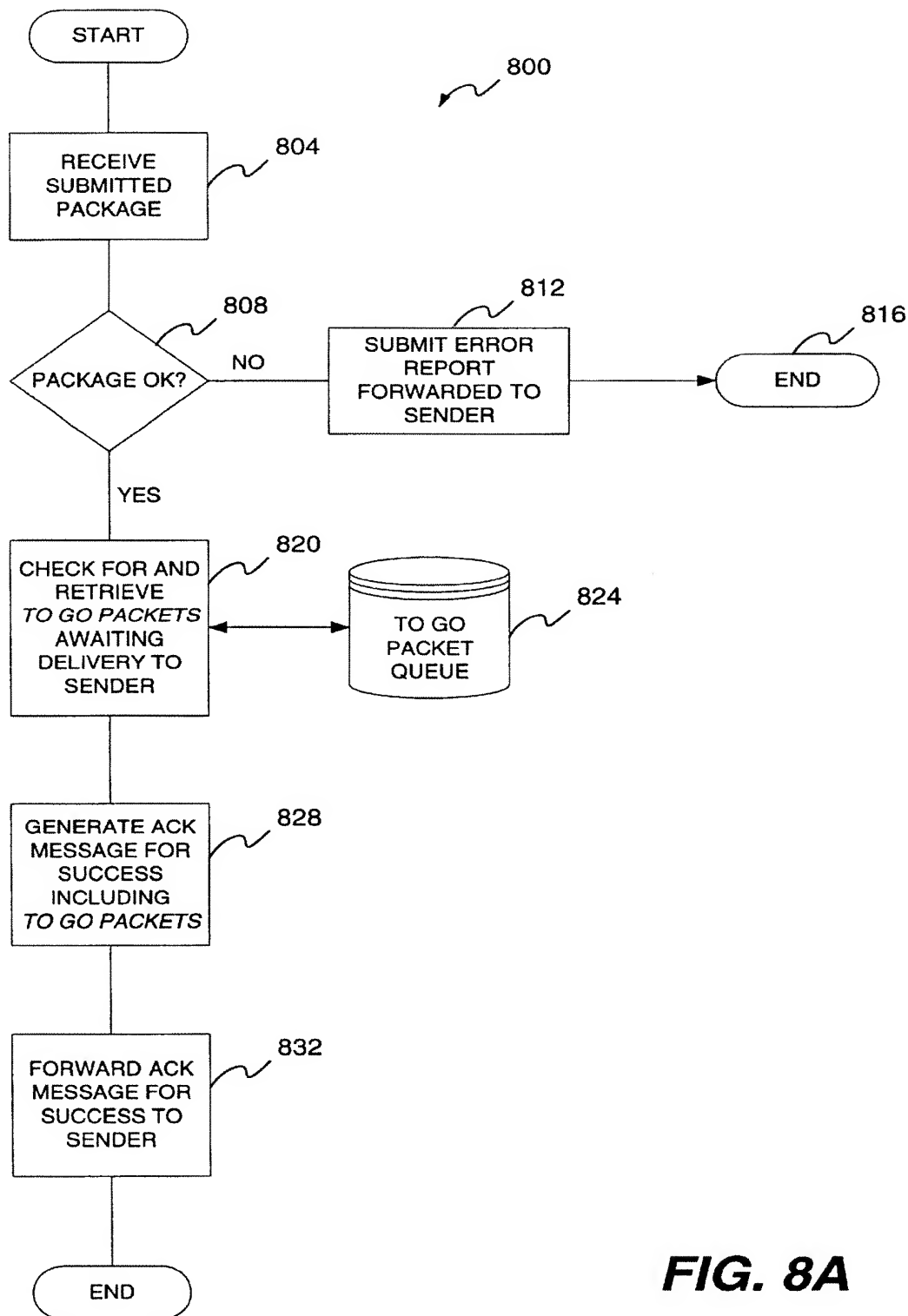
**FIG. 6B**

## DELIVERY SUCCESS REPORT

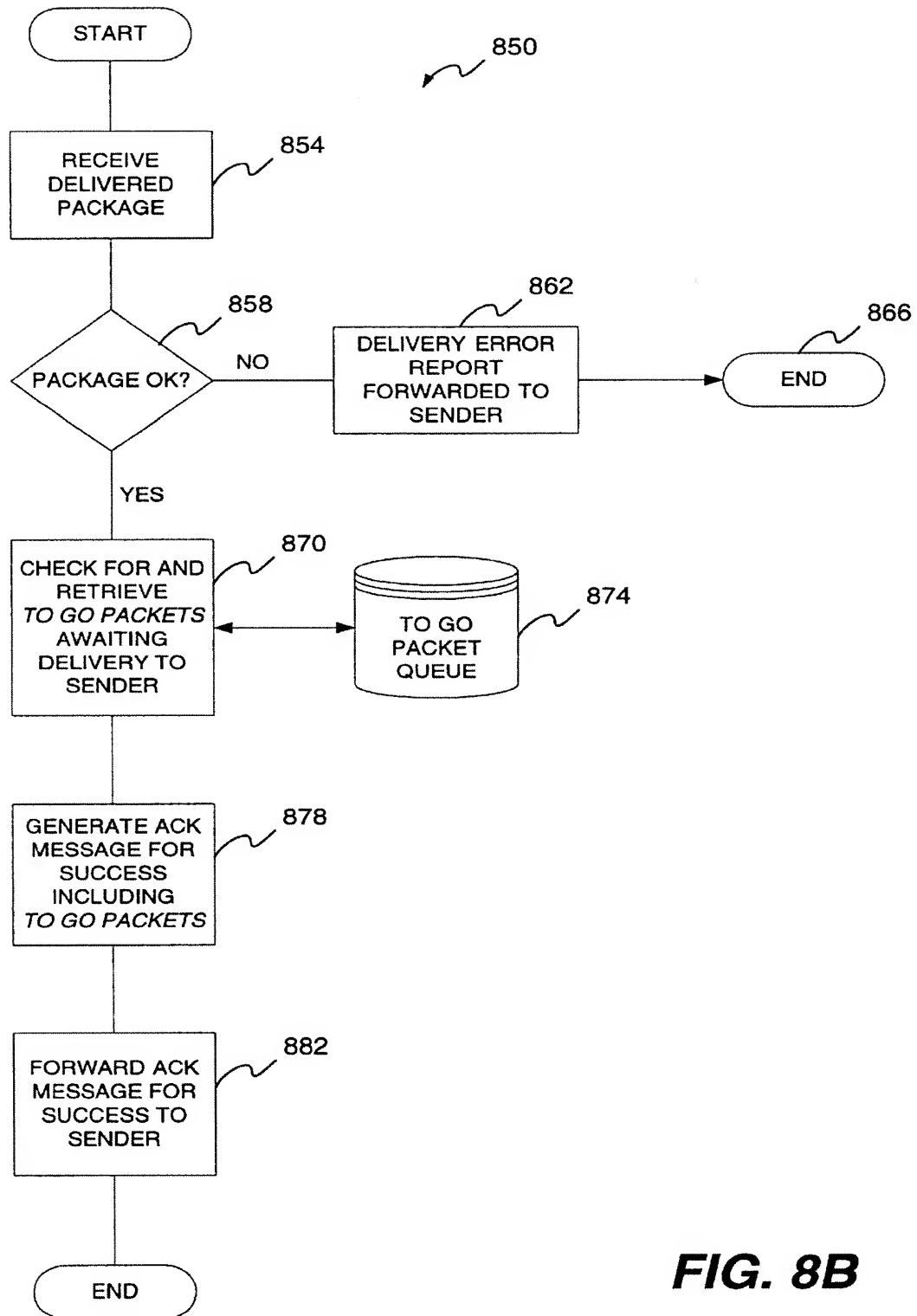
**FIG. 7A**

## DELIVERY ERROR REPORT

**FIG. 7B**

**FIG. 8A**



**FIG. 8B**

# SHORT MESSAGE SERVICE WITH IMPROVED UTILIZATION OF AVAILABLE BANDWIDTH

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to (i) U.S. application Ser. No. 09/170,879, filed Oct. 20, 1998, and entitled "WIRELESS MOBILE DEVICES HAVING IMPROVED OPERATION DURING NETWORK UNAVAILABILITY"; (ii) U.S. application Ser. No. 09/172,105, filed Oct. 13, 1998, and entitled "METHOD AND APPARATUS FOR PROVIDING ELECTRONIC MAIL SERVICES DURING NETWORK UNAVAILABILITY," now U.S. Pat. No. 6,289,212; (iii) U.S. application Ser. No. 08/977,572, filed Jul. 11, 1997, and entitled "PUSHING AND PULLING DATA IN NETWORKS," now U.S. Pat. No. 6,119,167; (iv) U.S. application Ser. No. 09/070,668, filed Apr. 30, 1998, and entitled "METHOD AND APPARATUS FOR PROVIDING NETWORK ACCESS OVER DIFFERENT WIRELESS NETWORKS," now U.S. Pat. No. 6,314,108; and (v) U.S. application Ser. No. 09/105,691, filed Jun. 26, 1998, and entitled "METHOD AND APPARATUS FOR FRAGMENTING MESSAGES FOR A WIRELESS NETWORK USING GROUP SHARING OF REFERENCE NUMBERS," now U.S. Pat. No. 6,185,208; the contents of all of which are hereby incorporated by reference.

## BACKGROUND OF THE INVENTION

### 1. Field of Invention

This invention generally relates to wireless communication systems and, in particular, to short message service (SMS) communications in wireless communication systems.

### 2. Discussion of Related Art

The tremendous growth of the Internet in recent years has fueled the need to provide wireless devices such as mobile telephones, personal digital assistants (PDAs) and the like with access to information and services available on the Internet. However, providing wireless devices with access to the Internet is complicated by the fact that various different carrier networks with different wireless network characteristics are used domestically and world wide to communicate with the wireless devices. Examples of wireless networks include Cellular Digital Packet Data (CDPD), Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA) to name a few, and each of these wireless networks has different data transfer characteristics such as latency, bandwidth, protocols and connection methods. As examples, protocols can be Internet Protocol (IP), Short Message System (SMS) and Unstructured Supplementary Service Data (USSD), and connection methods can include packet switched or circuit switched. A carrier transport ID indicates the protocol used by the network, such as User Datagram Protocol (UDP), Short Message Peer-to-peer Protocol (SMPP), or Wireless Datagram Protocol (WDP).

Wireless communications are used for voice and data communications. In the case of wireless data communications, one type of service that can be provided by wireless networks is SMS. Short Message Server Centers (SMSCs) associated with the wireless networks provide the SMS service. SMS gives subscribers the ability to receive a relatively small amount of information over the wireless networks. The information provided through SMS is generally referred to as messages and can, for example, include text messages, electronic mail (email), voice mail, message

alerts and pages from pagers. SMS tends to be a more cost effective means for the transmission of small amounts of data because SMS uses considerably less bandwidth than a typical wireless phone call or wideband network connection. SMS channel usage is also typically charged to subscribers at a fixed cost per month so its bandwidth, although limited, tends to be relatively inexpensive for subscribers.

One-way-SMS represents a narrowband channel that can carry data in primarily one direction, with acknowledgements going in the opposite direction. Two-way SMS allows bi-directional communications over SMS using a channel with a relatively low bandwidth, which is slightly greater in capacity than that provided by one-way SMS.

An SMS communications system can be thought of as a client-server type of system where a client device makes a request, and upon reception, a server device acknowledges whether or not the request was received in tact. In the case of SMS, the acknowledgements represent a success report if the request was successfully received or an error report if the request was not successfully received. For example, when a mobile device sends a message to an SMSC, the SMSC returns a report to the mobile device to either confirm receipt of the message or to notify of error in the delivery of the message. If the message is received successfully, the SMSC stores and forwards the message to an entity capable of receiving SMS messages. This forwarded message contains the address of the originating entity. In a similar fashion, when the SMSC delivers a message to a mobile device, the mobile device returns a report to the SMSC to either confirm receipt of the message or to notify of error in the delivery of the message.

These reports, which provide an indication of a successful or failed delivery process, are referred to as SMS acknowledgement messages. SMS acknowledgement messages are comprised of a plurality of pre-defined functional fields. Examples of successful acknowledgement reports include the Submit Success Report (SSR) and the Delivery Success Report (DSR). These multi-field success acknowledgement reports have a well-defined structure, which includes user data fields. These user data fields are generally not utilized at present and therefore they represent wasted bandwidth to the network. The failed reception acknowledgement error reports are referred to as error reports and do not have user data fields. SMS is further described in Global System for Mobile Communications (GSM) 03.40, versions 5.6.1, European Telecommunications Standards Institute (ETSI) (ETS 300 901), January 1998, which is hereby incorporated by reference.

Thus, given the growth of wireless services and the fixed cost pricing of SMS channels, there exists a need for more efficient utilization of SMS systems to accommodate an increase in subscribers and their usage.

## SUMMARY OF THE INVENTION

Broadly speaking, the present invention relates to techniques that enable wireless client devices to more efficiently utilize the available transmission bandwidth in a wireless network. In one embodiment, the invention operates to include or incorporate return information (data) in an acknowledgement message after an incoming message has been successfully received from a sender.

The invention can be implemented in numerous ways, including as a method, an apparatus, and a computer readable medium. Several embodiments of the invention are discussed below.

As a method for sending messages between a client device and a server device through a narrowband channel of

3

a wireless data network, one embodiment of the invention includes the acts of: receiving a message at the client device, the message being sent from the server device to the client device through the narrowband channel of the wireless data network; preparing an acknowledgement message to be returned to the server device, the acknowledgement message including at least a portion of another message destined for the server device; and sending the acknowledgement message to the server device. As an example, the client device can be a personal digital assistant, a mobile telephone device, or a pager.

As a method of transmitting message packets from an initiating unit to a receiving unit over a wireless data network using a Short Message Service Center, one embodiment of the invention includes the acts of: maintaining, at the receiving unit, a message queue of messages awaiting delivery; receiving, at the receiving device, a message from the initiating unit over the wireless communications using the Short Message Service Center; determining whether the received message is valid; determining whether the message queue contains a deferred message awaiting delivery to the initiating unit; generating an acknowledgement message that incorporates at least a portion of the deferred message awaiting delivery to the initiating unit; and forwarding the acknowledgement message to the wireless client device over the wireless communications using the Short Message Service Center.

As a computer readable medium including computer program code for sending messages between a client device and a server device through a channel of a wireless data network, one embodiment of the invention includes: computer program code for receiving a message at the client device, the message being sent from the server device to the client device through the channel of the wireless data network; computer program code for preparing an acknowledgement message to be returned to the server device, the acknowledgement message including data destined for the server device; and computer program code for sending the acknowledgement message to the server device.

As a computer readable medium including computer program code for of transmitting message packets from an initiating unit to a receiving unit over a wireless data network using a Short Message Service Center, one embodiment of the invention include: computer program code for maintaining, at the receiving unit, a message queue of messages awaiting delivery; computer program code for receiving, at the receiving device, a message from the initiating unit over the wireless communications using the Short Message Service Center; computer program code for determining whether the received message is valid; computer program code for determining whether the message queue contains a deferred message awaiting delivery to the initiating unit; computer program code for generating an acknowledgement message that incorporates at least a portion of the deferred message awaiting delivery to the initiating unit; and computer program code for forwarding the acknowledgement message to the wireless client device over the wireless communications using the Short Message Service Center.

As an apparatus for sending and receiving messages over a wireless data network, one embodiment of the invention includes: an outgoing data queue that stores data to be sent over the wireless data network; a message manager, the message manager manages (i) the reception of incoming messages from senders over the wireless data network and (ii) the generation of outgoing messages to be sent over the wireless data network; a storage medium that stores the

4

incoming messages; and a processing module operatively connected to said message manager and said storage medium, said processing module interacts with said storage medium and said message manager to request, send and receive data over the wireless data network. The outgoing messages generated by said message manager include acknowledgement messages that acknowledge the receipt of at least some of the incoming messages. Depending on availability of data in said outgoing data queue, the acknowledgement messages generated by said message manager include data from said outgoing data queue destined for the respective senders of the incoming messages being acknowledged.

The advantages of the invention are numerous. Different embodiments or implementations may yield one or more of the following advantages. One advantage of the invention is that wireless devices to more efficiently utilize the available transmission bandwidth of a narrowband channel (e.g., SMS channel) in a wireless network. Another advantage of the invention is that it facilitates cost-effective use of a narrowband channel (e.g., SMS channel) in a wireless network. Still another advantage of the invention is that non-time critical messages can be sent over a wireless network in efficient, cost-effective way.

Other aspects and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying drawings which illustrate, by way of example, the principles of the invention.

#### BRIEF DESCRIPTION OF DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein the reference numerals illustrate the structural elements, and in which:

FIG. 1A is a block diagram of a wireless communication system according to one embodiment of the invention;

FIG. 1B is a block diagram of a communication system according to one embodiment of the invention;

FIG. 1C is a flow diagram of a message acknowledgement processing according to an embodiment of the invention;

FIG. 2A is a block diagram of a wireless data communication system according to another embodiment of the invention;

FIG. 2B is a block diagram of a wireless client device according to one embodiment of the invention;

FIG. 3 is a block diagram of proxy server device according to one embodiment of the invention;

FIG. 4 illustrates a functional block diagram of Short Message Service (SMS) server according to one embodiment of the invention;

FIG. 5 illustrates a functional block diagram of the client-server relationship between a wireless client device and a Short Message Service (SMS) server;

FIG. 6A illustrates a format for a Submit Success Report (SSR);

FIG. 6B illustrates a format for a Submit Error Report (SER);

FIG. 7A illustrates a format for a Delivery Success Report (DSR);

FIG. 7B illustrates a Delivery Error Report (DER);

FIG. 8A is a flow diagram of Submit Success Report (SSR) processing according to an embodiment of the invention; and

FIG. 8B is a flow diagram of Delivery Success Report (DSR) processing according to an embodiment of the invention.

## 5

DETAILED DESCRIPTION OF THE  
INVENTION

The present invention relates to techniques that enable wireless client devices to more efficiently utilize the available transmission bandwidth in a wireless network. In one embodiment, the invention operates to include or incorporate return information (data) in an acknowledgement message after an incoming message has been successfully received from a sender.

The invention is particularly applicable to a Global System for Mobile Communications (GSM) network that is capable of bi-directional communications with a short message service center (SMSC). It will be appreciated by one of ordinary skill in the art that invention could be applied to wireless networks other than such GSM networks.

Wireless client devices, also referred to as mobile devices or wireless communication devices, include but are not limited to personal digital assistants (PDAs), mobile telephones (including cellular phones), pagers, or wireless capable remote controllers. Typically, these wireless client devices have much less computing resource than a desktop or laptop computer does.

In the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will become obvious to those skilled in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuitry have not been described in detail to avoid unnecessarily obscuring aspects of the present invention.

Embodiments of the invention are discussed below with reference to FIGS. 1A–8B. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments.

FIG. 1A is a block diagram of a wireless communication system 10 according to one embodiment of the invention. Wireless communication system 10 supports a plurality of wireless communication devices (also known as mobile devices, wireless client devices, etc.). For convenience, only a single wireless communication device 12 is illustrated in FIG. 1A. Wireless communication device 12 communicates with a network gateway 14 through a wireless network 16. Also, wireless communication device 12 communicates with a Short Message Service Center (SMSC) 18 through wireless network 16. SMSC 18 also connects to network gateway 14. Often, a first carrier network in wireless network 16 provides a wideband channel that couples network gateway 14 to wireless communication device 12, and a second carrier network in wireless network 16 provides a narrowband channel that couples SMSC 18 to wireless communication device 12. Subscribers that use the plurality of wireless communication devices are normally charged for connect time when using the wideband channel. However, use of the narrowband channel is often charged to subscribers on a fixed monthly rate.

Network gateway 14 couples to a wired network 20. Wired network 20 is a data network that interconnects various remote devices, including a remote device 22 illustrated in FIG. 1A. Remote device 22 can, for example, be a server machine or a client machine. Wired network 20 can pertain to the Internet, an Intranet, or some other data network.

Wireless communication system 10 enables wireless communication device 12 to send and receive messages from the

## 6

remote device 22. Such messages traverse wireless network 16, wired network 20, and either the narrowband or wideband channel. Typically, messages destined for the narrowband channel are relatively short in length. The messages destined for the narrowband channel are often known as SMS messages or short messages and have a pre-determined maximum size (e.g., 140 bytes). However, a process known as fragmentation allows messages greater than the maximum size to be sent over the narrowband channel. On the other hand, larger messages are often sent over the wideband channel at higher speed but at a greater cost.

According to the invention, the narrowband channel is more efficiently utilized to take advantage of this inexpensive resource within wireless communication system 10. More particularly, the invention manages the sending of messages over the narrowband channel such that acknowledgement messages are able to carry additional data from recipients of the messages to their senders with reduced overhead. The reduction in overhead results because fewer messages need to be generated and thus, less of the bandwidth of the narrowband channels has to be allocated for transmission of messages. The reason that messages need to be generated and transmitted is because previously unused bandwidth (e.g. the user data fields) are utilized to carry the additional data. The additional data can be related or unrelated to the acknowledgement message that carries the additional data. One example of related data is a reply to the message such as a resource returned to the sender.

FIG. 1B is a block diagram of a communication system 30 according to one embodiment of the invention. In this embodiment, communication system 30 utilizes a SMSC (or SMC server) between a mobile device and a remote device. Typically, the remote device is coupled to the SMSC through a network gateway (e.g., proxy server) such as shown in FIG. 1A. Although the network gateway is advantageous for management and protocol conversion operations, a communication system according to the invention could also operate without a network gateway. The SMSC can also couple directly to a wired network (e.g., the wired network) and/or the remote device.

Communication system 30 illustrated in FIG. 1B includes a mobile device 32 and a SMSC 34. The communication between mobile device 32 and SMSC 34 is through a wireless communication channel of a wireless network. Typically, the wireless communication channel is a narrowband channel. As an example, the wireless communication channel can be a SMS channel. Among other things, mobile device 32 includes a client module 36 (e.g., a network browser), a message receive manager 38, a storage area 40, a message send manager 42, and an outgoing queue 44.

Mobile device 32 communicates with SMSC 34 to obtain resources from a remote server. The remote device contains information or resources that mobile device 32 may desire. Initially, the client module 36 (e.g., network browser) requests a resource that originates on the remote device. However, the information or resource often need not be immediately provided and thus can be performed in a non-time critical manner. The client module 36 making a request does not wait to receive the resource; instead, the resource is to be acquired independent of further operation of the client module 36. As an example, such asynchronous requests are particularly useful in certain situations such as where a remote server needs to eventually be updated with some event or action that has occurred on mobile device 32, but mobile device 32 does not need to wait until the remote server is updated before it continues.

Thus, where the request is non-time critical, the client module 36 will forward a request message to the outgoing

message queue 44. From the perspective of the client module 36, once the request message has been successfully queued, processing by the client module 36 continues. Then, the message send manager 42 will manage the retrieval of messages (e.g., request messages) from the outgoing message queue 44 and their delivery to the SMSC 34 over the wireless communication channel. The message send manager 42 can also receive time critical requests (or high priority request messages) directly from the client module 36.

At SMSC 34 the incoming messages are received by a message receive manager 44 and temporarily stored in a storage area 46. A server module 48 at the SMSC 34 operates to service the incoming messages. The server module 48 forwards the incoming messages (e.g., request messages) from the storage area 46 to the appropriate remote devices over a network link 50. Messages (whether reply messages or request messages) from the remote device are directed through the server module 48 to a message send manager 52 or to an outgoing message queue 54. Typically, those messages being sent to the outgoing message queue 54 are non-time critical messages. The message send manager 52 manages the sending of the messages received from the server module 48 or from the outgoing message queue 54 over the wireless communication channel to the mobile device 32.

Normally, whenever a message is received, it is acknowledged. When message send manager 52 is preparing to send an acknowledgement message over the wireless communication channel to a particular mobile device as is typically done to acknowledge receipt of a message, message send manager 52 examines outgoing message queue 54 to determine if any messages therein are also destined for the particular mobile device. When there is a message in the outgoing message queue 54 for the particular mobile device, then the message (or a portion thereof) is inserted into the acknowledgement message and thus sent with the acknowledgement. Periodically, the message send manager 52 can review the contents of the outgoing message queue 54 to insure no messages are delayed too long before being sent.

When the wireless communication channel through the wireless network is available, message send manager 52 operates to send messages to the message receive manager 38 of mobile device 32 via the wireless communication channel. Message receive manager 38 then supplies the incoming messages to storage area 40. Message receive manager 38 can also notify client module 36 that the requested resource has been received. When the incoming message is an acknowledgement message, message receive manager 38 is able to parse the acknowledgement to separate the additionally inserted message (or portion thereof) from the acknowledgement message.

The information can be exchanged in either or both directions. More particularly, the use of acknowledgement messages also works in the reverse direction—for acknowledgement messages sent from mobile device 32 or to SMSC 34. These acknowledgement messages can carry information or resources from mobile device 32 to SMSC 34 or some remote device. SMSC 34 typically couples to a network gateway which in turn couples to a remote server. Upon receiving the acknowledgement message, server module 48 forwards the acknowledgement message to the initial sender. Typically, the initial sender of the message being acknowledged is a remote device coupled to the SMSC 34 via a network gateway. As needed, the acknowledgement message can be modified to produce a status message for transport to the initial sender. The status message is also

considered an acknowledgement message even though possibly modified for other network characteristics (e.g., protocols).

Further, any device receiving messages is able to process an acknowledgement message to include additional information as discussed above. Although FIG. 1B indicates that the mobile device 32 and the SMSC 34 can operate to provide the more efficient use of acknowledgement messages. Additionally, either or both the network gateway and the remote device can also implement the more efficient use of acknowledgement messages.

In general, the destination (recipient) device for the initial message needs to be able to process an acknowledgement message to include other information destined for the sender device (i.e., same destination address). The destination device (mobile, SMSC, network gateway or remote device) for the modified acknowledgement message needs to understand how to parse the modified acknowledgement message. A message typically includes a source address, a destination address, and data. The data can take a variety of forms and have a variety of effects. The data can, for example, request information from a destination device, supply information to a destination device, cause some action at a destination device.

FIG. 1C is a flow diagram of a message acknowledgement processing procedure 60 according to an embodiment of the invention. Message acknowledgement processing procedure 60 is performed by any device or server that operates to acknowledge receipt of messages. For example, message acknowledgement processing procedure 60 can be performed by mobile device 32 or SMSC 34 illustrated in FIG. 1B or network gateway 14, SMSC 18 or remote device 22 illustrated in FIG. 1C. In this embodiment, the messages are considered to be SMS messages, which are messages of a limited size. However, other types of messages besides SMS messages can also be used.

Message acknowledgment processing procedure 60 begins with a decision block 62 that determines whether a SMS message has been received. When decision block 62 determines that a SMS message has not yet been received, message acknowledgment processing procedure 60 awaits the reception of such a message. Once a SMS message has been received, message acknowledgment processing 60 is effectively invoked.

After message acknowledgment processing procedure 60 is invoked, the SMS message is stored at block 64. Here, the SMS message is stored, for example, in an incoming message queue. Then, a decision block 66 determines whether the SMS message is valid. If the decision block 66 determines that the SMS message is not valid, then an error message is prepared at block 68. Then, the error message is sent at block 70. The error message is sent to the sender of the SMS message that was received at block 62. Representative error messages for SMS are described below with reference to FIGS. 6B and 7B. Following block 70, the message acknowledgment processing 60 is complete and ends when the received SMS message was not valid.

On the other hand, when decision block 66 determines that the SMS message is valid, then the processing module is notified of the SMS message received at block 72. As an example, the processing module can be the client module 36 of the mobile device 32, the server module 48 of the SMSC 34, or some other processing module of another device (e.g., proxy server or remote server). Then, an SMS acknowledgment message is prepared at block 74. At this point, the SMS acknowledgment message resembles a conventional

acknowledgment message typically automatically produced by the server that has received the SMS message. Representative acknowledgement messages for SMS are described below with reference to FIGS. 6A and 7A.

Next, a decision block 76 determines whether an outgoing message queue is empty. When decision block 76 determines that the outgoing message queue is not empty, then data obtained from the outgoing message queue is added to the SMS acknowledgment message at block 78. The data from the outgoing message queue that is added to the SMS acknowledgement message is data that is destined for the original sender of the received SMS message.

On the other hand, when decision block 76 determines that the outgoing message queue is empty, block 78 is bypassed. Following block 78, as well as following the decision block 76 when the outgoing message queue is empty, the SMS acknowledgment message is then sent at block 80. The SMS acknowledgment message is sent to the original sender of the SMS message that was received. Following block 80, message acknowledgment processing procedure 60 is complete and ends.

Here, it should be noted that the SMS acknowledgement message is automatically directed back to the sender of the received SMS message being acknowledged. In some embodiments, the SMS acknowledgement message is directed back to an intermediate server and then directed back again to the initial sender of the message. For example, where a message originated by an initiating machine and sent to a SMSC which in turn sends a SMS message to the receiving machine, the SMS acknowledgement is first sent to the SMSC and then the SMSC can send an acknowledgement to the initiating machine. Hence, the acknowledgement message is able to carry the additional data back to the initiating machine.

In one implementation, the SMS acknowledgment message has a user data field. The data from the outgoing message queue is added to the user data field while updating another field known as the user data field length. Typically, the user data field is limited in size so typically only a limited amount of data from the outgoing message queue would be added to the SMS acknowledgment message. In one example, the user data field can carry all the data for a particular message stored in the outgoing message queue. However, in more sophisticated designs, it is possible that multiple relatively short messages from the outgoing message queue could be provided in the user data field such that it carries a plurality of messages. Still further, the messages from the outgoing message queue could be fragmented before being added into the user data field to allow for greater length messages to be sent in this manner over multiple acknowledgement messages.

Also, it should be noted that the data added to the acknowledgement messages need not be traditional text messages but could be any type of data that might be exchanged between sending and receiving devices. For example, the data could be a text message, an indicator for a voice mail message, electronic mail, a page, a resource (previously requested or being pushed to the recipient), an alert, a configuration file, a script, or an executable.

FIG. 2A is a block diagram of a wireless data communication system according to another embodiment of the invention. The wireless data communication enables wireless client devices, including a representative wireless client device 100, to communicate with information servers over one or more networks. As examples, wireless client device 100 can be a mobile phone, a cellular phone, a palm sized computing device or a personal digital assistant.

Wireless client device 100 is coupled to a proxy server device 140 and a short message service (SMS) server 150 through an airnet (e.g., a wireless network) 120. Airnet 120 can, for example, be a GSM network. Proxy server device 140 is also known as a gateway server, a link server, or a network gateway. Proxy server device 140 acts as a bridge between landnet 160 and airnet 120 with respect to messages requiring larger bandwidths. Short message service (SMS) server 150 also acts as a bridge between the proxy server device 140 and the wireless client device 100 for messages requiring smaller bandwidths (i.e., short messages).

Proxy server device 140 is further coupled to a landnet 160 to which a personal computer 170 and an information server 180 are coupled. The landnet 160 is a land-based, wired network that may include the Internet, an Intranet, or some other wired data network. Information service server 180 is representative of one or more network servers coupled to landnet 160 and providing hypermedia information including mobile data information for wireless client device 100. The personal computer 170 represents one or more standard computers (desktops or laptops) that can be used to send messages to wireless client device 100.

In addition, information service server 190 is representative of one or more network servers that can couple directly to SMS server 150 via a direct interface 192 (e.g., X.25) or a private network. The information service server 190 is representative of one or more network servers coupled directly to short message service (SMS) server 150 via the direct interface (e.g., X.25) 192. Both the information servers 180 and 190 can be used to provide hypermedia information for the wireless client device 100. The proxy server device 140, the short message service (SMS) server 150, and information service servers 180 and 190 are typically operated on workstation computers utilizing an operating system suitable for networked environments such as Microsoft Windows NT or Unix.

Airnet 120 is a wireless network and an antenna 121 represents a wireless carrier infrastructure for airnet 120. The wireless carrier infrastructure generally includes a Base Station Subsystem (BSS), a Network Subsystem (NSS) and an operations and maintenance center. The base station controls radio or telecommunication links with wireless client device 100. The operations and maintenance center can include a mobile switching center performing the switching of calls between the mobile devices and other fixed or mobile network users. Further, the operations and maintenance center manages mobile account services, such as authentication, and oversees the proper operation and setup of the wireless network. Each of the components and processes of the wireless carrier infrastructure are known to those skilled in the art and are not described herein to avoid unnecessarily obscuring the aspects of the invention.

According to one embodiment, the communication protocol in the landnet 160 is HyperText Transfer Protocol (HTTP) (or HTTPS a secure version of HTTP) and runs on Transmission Control Protocol (TCP). According to one embodiment, the upper level communication protocol in airnet 120 is Wireless Access Protocol (WAP) or Handheld Device Transport Protocol (HDTP) (formerly known as Secure Uplink Gateway Protocol (SUGP)), which preferably run on Wireless Datagram Protocol (WDP) or User Datagram Protocol (UDP). The above-described protocols have been provided for purposes of illustration and not restriction. One of ordinary skill in the art will appreciate that the present invention can be practiced using other land based and wireless protocols.

FIG. 2B is a block diagram of a wireless client device 200 according to one embodiment of the invention. Wireless



11

client device 200 can, for example, represent wireless client device 100 of FIG. 2A.

Wireless client device 200 includes a processor 202 and a client module 204. Client module 204 includes processes that are performed by processor 202 to operate wireless client device 200.

Client module 204 works in conjunction with processor 202 and a working memory 212 to perform the processing tasks performed by wireless client device 200 including establishing a communication session with proxy server device 140 via airnet 220, requesting and receiving data via airnet 220, displaying information on the display 208, and receiving user (subscriber) input from a user via the keypad 206. Client module 204 can also operate, among other things, implement a browser, commonly referred to as micro-browser, which requires much less computing power and memory than well-known HTML browsers do. The micro-browser is, preferably, a HDML micro-browser. One such micro-browser is, for example, available from Unwired Planet, Inc. located at 800 Chesapeake Drive, Redwood City, Calif. 94063. Additional details on accessing a server device from a mobile device using a micro-browser are described in U.S. Pat. No. 5,809,415, which is hereby incorporated by reference.

Wireless client device 200 includes a WCP interface 214 that couples to airnet (wireless network) 220 via a radio-frequency (RF) transceiver (not shown) to receive incoming and outgoing signals over a wireless channel 232. A device identifier (ID) storage 216 supplies a device ID to WCP interface 214. The device ID identifies a specific code that is associated with wireless client device 200. The device ID is used by proxy server device 140 to associate wireless client device 200 with a user account provided in proxy server device 140. The device ID can be a phone number of the device or a combination of an IP address and a port number. An example of a combination of an IP address and a port number is 204.163.165.132:01905 where 204.163.165.132 is the IP address and 01905 is the port number. The device ID is further associated with a subscriber ID authorized by a wireless network carrier (and stored in proxy server device 140) as part of the procedures to activate a subscriber account for wireless client device 200. The subscriber ID is a unique identification to a subscriber of wireless client device 200. In other words, each of the wireless client devices serviced by proxy server device 140 has a unique device ID that corresponds to a respective user account in proxy server device 140.

Wireless client device 200 also includes voice circuitry 218 (e.g., a speaker and a microphone) and the associated hardware (e.g., an encoder/decoder 210, processor 202 and the keypad circuitry 206) which provide a telephone mode of operation which is separate and distinct from a data mode of operation used when interfacing with proxy server device 140. In the telephone mode of operation, a subscriber can cause wireless client device 200 to place a phone call to another party having a phone, either wireless or land-based.

A message queue (e.g., outgoing message queue) can be maintained in the working memory 212 and managed by the client module 204. The client module 204 and the processor 202 also implement a message manager (i.e., message receive manager 38 and message send manager 42).

FIG. 3 is a block diagram of proxy server device 340 according to one embodiment of the invention. Proxy server device 340 can, for example, be the proxy server device 140 of FIG. 2A. Proxy server device 340 serves as a gateway between the airnet 120 and landnet 160.

12

Proxy server device 340 includes a Land Control Protocol (LCP) interface 358 that couples to landnet 160, and a Wireless Control Protocol (WCP) interface 341 that couples to airnet 120. A server module 343 is coupled between the LCP interface 358 and the WCP interface 341. Further, it will be appreciated that the principles of the invention can be used with a wide variety of wireless networks, including such wireless networks as CDPD, GSM, CDMA and TDMA, to name a few.

Server module 343 performs traditional server processing as well as protocol conversion processing from one communication protocol to another communication protocol. According to one embodiment, the protocol conversion processing can be implemented as a separate module referred to as a mapper. In the case of protocol conversion between HDTP (or WSP) and HTTP, the conversion is a straight data mapping relationship. It is understood to those skilled in the art that WCP interface 341 can be readily replaced by other interface module when airnet 120 uses other communication protocol, the same is true to LCP interface 358 when landnet 160 uses a different communication protocol.

Server module 343 includes a control engine 344 and a message processor 357. Control engine 344 interacts with SMS server 150 via WCP interface 341 to coordinate the reception or delivery of messages (including notifications and requests). The message processor 357 receives messages from landnet 160 via LCP interface 358 and performs a series of processing and management activities.

Server module 343 also includes an account manager 348 and an account interface 346. Account manager 348 manages a plurality of user accounts for all the mobile devices serviced by proxy server device 340. Each of the wireless client devices 100 serviced by proxy server device 340 may be assigned a device identification (ID). Account manager 348 is responsible for creating a user account for each of the wireless client devices that communicate with proxy server device 340. Account manager 348 control access of wireless client devices to services provided by the proxy server device 340 and the SMS server 150.

It is understood that the user accounts may be stored in another network server coupled proxy server device 340. In other words, the user accounts can be kept in a database that is physically placed in any computing device coupled to landnet 160.

Proxy server device 340 also includes a processor 356a and storage 356b as the primary hardware components. Processor 356a performs operations in accordance with the server module 343. It should be understood to those skilled in the art that the proxy server device 340 is a piece of hardware equipment that includes one or more processors (e.g., processor 356a), working memory (e.g., storage 356b), buses, interfaces and other components. On the other hand, the server module 210 represents a software module, which contains processes (e.g., compiled and linked processes) loaded into the working memory to perform designated functions by proxy server device 340. The same distinction is equally applied to client modules within the wireless client devices.

FIG. 4 illustrates a functional block diagram of Short Message Service (SMS) server 400 according to one embodiment of the invention. SMS server 400 can, for example, be SMS server 150 of FIG. 2A.

SMS server 400 includes a Short Message Service (SMS) kernel 440 coupled between two Wireless Control Protocol (WCP) interfaces 404 and 418, a processor 410, a storage

13

device 414, and an X.25 interface 422. Storage device 414 can store databases and messages for the serviced wireless client devices. For example, SMS server 400 can utilize storage device 414 to store an outgoing queue of data (messages) awaiting delivery for each of the wireless client devices being supported by SMS server 400.

SMS kernel 440 is typically loaded in memory and executed by processor 410 to perform traditional server processing. SMS kernel 440 can also perform protocol conversion from one communication protocol to another communication protocol (i.e., for the X.25 interface). The X.25 interface is an often used with public data communications networks. X.25 interface 422 provides a connection with information server 190 without having to go through proxy server device 140.

SMS kernel 440 includes a system manager 444, an overall administration and maintenance (OA&M) module 448, a gateway module 452, an inter-working function module 456, a message/alert management module 460, and a database management application 464. System manager 444 manages overall system operation. OA&M module 448 provides services for billing, administration and network management. Gateway module 452 coordinates activity between proxy server device 140 and SMS server 400. Inter-working function module coordinates activity between SMS server 400 and information server 190, such as through X.25 interface 422. Message/alert management module 460 provides management for transmission and reception of messages and providing of alerts. The database management application 464 manages a database of user account information.

SMS server 400 is coupled to proxy server device 140 through a land-based channel 468. SMS server 400 is coupled to the wireless client devices it services through airnet 120. In this embodiment, the communication protocol used in airnet 120 and over land-based channel 468 is WAP or HDTP, which preferably runs on UDP or WDP. SMS server 400 is also coupled to the information server 190 through the X.25 interface 422. The protocol for the X.25 interface 422 can be any protocol supported by the X.25 standard.

FIG. 5 illustrates a functional block diagram of the client-server relationship between a wireless client device 500 and a Short Message Service (SMS) server 510. As an example, wireless client device 500 and SMS server 510 may represent wireless client device 100 and SMS server 150 of FIG. 2A, respectively. According to one scenario, during its operation, wireless client device 500 will submit a request 502 to SMS server 510. Typically, in the case of SMS, the request is a short message that is to be directed to an addressee. SMS server 510 receives the request from wireless client device 500 and determines whether the request is valid. For example, a request can be deemed valid when it does not have errors, has not expired, and the message has not been received before. When the request is determined to be valid, an acknowledgement 504 is sent from SMS server 510 to wireless client device 500. Acknowledgement 504 is typically done with an acknowledgement (ACK) message such as a Submit Success Report (SSR). On the other hand, when the request is determined not to be valid, then an error notification is sent from SMS server 510 to wireless client device 500. As an example, the error notification is an error message such as a Submit Error Report.

In a similar fashion, according to another scenario, when SMS server 510 has information to be delivered to wireless

14

client device 500, the information is transmitted 506 to wireless client device 500. Typically, the information is contained in a short message to be delivered to wireless client device 500. Upon receiving the information, wireless client device 500 determines if the received information is valid. For example, a request can be deemed valid when it does not have errors, has not expired, and the message has not been received before. When the information is determined to be valid, a delivery acknowledgement 508 is sent from the wireless client device 500 to the SMS server 510. Delivery acknowledgement 508 is typically done with an acknowledgement (ACK) message such as a Delivery Success Report (SSR). On the other hand, when the information is determined not to be valid, then an error notification is sent from wireless client device 500 to SMS server 510. As an example, the error notification is an error message such as a Delivery Submit Error Report (DER).

In the case of SMS, the Submit Success Report (SSR), Submit Error Report (SER), the Delivery Success Report (DSR), and Delivery Error Report (DER) have well-defined structures as described with reference to FIGS. 6A, 6B, 7A and 7B. Additional details pertaining to these structures are provided in Global System for Mobile Communications (GSM) 03.40, versions 5.6.1, European Telecommunications Standards Institute (ETSI) (ETSI 300 901), January 1998, which has been previously incorporated by reference.

FIG. 6A illustrates a format for a Submit Success Report (SSR). The Submit Success Report (SSR) includes a message type indicator (TP-MTI) 600, a User-Data-Header Indication (TP-UDHI) 604, an Optional Parameter Indicator (TP-PI) 608, a SMS server time stamp (TP-SCTS) 612, a Protocol Identifier (TP-PID) 616, a Data Coding Scheme (TP-DCS) 620, a User Data Length indicator (TP-UDL) 624, and User Data (TP-UD) 628. User Data (TP-UD) 628 may include a User Data Header (UDH) comprised of reference number, an index indicating the total number of chunks of user data, and a chunk index. User Data (TP-UD) 628 is ordinarily unused in the SSR acknowledgements.

FIG. 6B illustrates a format for a Submit Error Report (SER). The Submit Error Report (SER) includes a message type indicator (TP-MTI) 632 and a Failure Cause (TPFCS) 636.

FIG. 7A illustrates a format for a Delivery Success Report (DSR). The Delivery Success Report (DSR) includes a message type indicator (TP-MTI) 700, a User-Data-Header Indication (TP-UDHI) 704, an Optional Parameter Indicator (TP-PI) 708, a Protocol Identifier (TP-PID) 712, a Data Coding Scheme (TP-DCS) 716, a User Data Length Indicator (TP-UDL) 720, and User Data (TP-UD) 724. The User Data 724 may include a User Data Header (UDH) comprised of a reference number, an index indicating the total number of chunks of user data, and a chunk index. User Data (TP-UD) 724 is ordinarily unused in the DSR acknowledgements.

FIG. 7B illustrates a Delivery Error Report (DER). The Delivery Error Report (DER) includes a Message Type Indicator (TP-MTI) 728 and a Failure Cause (TPFCS) 732.

FIG. 8A is a flow diagram of Submit Success Report (SSR) processing 800 according to an embodiment of the invention. A Short Message Service (SMS) server (e.g. 510 of FIG. 5) receives 804 a submitted request packet from a wireless client device (e.g. 500 of FIG. 5). The SMS server makes a determination 808 regarding the validity of the received packet. If the received packet is determined to be invalid a Submit Error Report (SER) (FIG. 6B) is generated and forwarded 812 to the originating wireless client device



15

816. If the received packet is determined to be valid, a check is made for packets awaiting delivery to the sender (TO GO PACKETS). These packets are stored in the TO GO PACKET storage queue 824. This process may take place in the SMS server or in any server with a connection to the SMS server. The TO GO PACKETS awaiting delivery to the sender are retrieved 820. When the Submit Success Report (SSR) is generated 828, the TO GO PACKETS are inserted as user data (TP-UD) and the User Data Length Indicator (TP-UDL) is updated to reflect the insertion. Finally, the Submit Success Report (SSR) including the TO GO PACKETS are forwarded 832 to the originating wireless client device. Following the forwarding 832, SSR processing 800 is complete and ends.

FIG. 8B is a flow diagram of Delivery Success Report (DSR) processing 850 according to an embodiment of the invention. A wireless client device (e.g. 500 of FIG. 5) receives 854 an information packet from an originating server, which may be the SMS server or any server connected to the SMS server. The wireless client device makes a determination 858 regarding the validity of the received information packet. If the received information packet is determined to be invalid, a Delivery Error Report (DER) (FIG. 7B) is generated and forwarded 862 to the originating server. The originating server can be the SMS server or any server connected to the SMS Server. If the received packet is determined to be valid, a check is made for packets awaiting delivery to the sender (TO GO PACKETS). These packets are stored in the TO GO PACKET storage queue 874 resident on the wireless client device. The TO GO PACKETS awaiting delivery to the sender are retrieved 870. When the Delivery Success Report (DSR) is generated 878, the TO GO PACKETS are inserted as user data field (TP-UD) and the User Data Length Indicator (TP-UDL) is updated to reflect the insertion. Finally, a Delivery Success Report (DSR) including the TO GO PACKETS are forwarded 882 to the originating server. Following the forwarding 882, the DSR processing 850 is complete and ends.

In the case of both the Submit Success Report (SSR) and the Delivery Success Report (DSR) the User Data Field (TP-UD) has an associated User Data Length Indicator (TP-UDL). As previously stated, the User Data (TP-UD) may contain a UDH comprised of a reference number, an index indicating the total number of chunks of user data, and a chunk index. This structure provides a mechanism to parse (or fragment) large messages (larger than the space provided in TP-UD) and send them as smaller chunks of information (chunk encoding).

The advantages of the invention are numerous. Different embodiments or implementations may yield one or more of the following advantages. One advantage of the invention is that wireless devices to more efficiently utilize the available transmission bandwidth of a narrowband channel (e.g., SMS channel) in a wireless network. Another advantage of the invention is that it facilitates cost-effective use of a narrowband channel (e.g., SMS channel) in a wireless network. Still another advantage of the invention is that non-time critical messages can be sent over a wireless network in efficient, cost-effective way.

While only certain embodiments of the invention have been illustrated and described herein, many modifications, substitutions, changes, and equivalents will now occur to those skilled in the art. It is therefore to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the invention.

16

What is claimed is:

1. A method for sending messages between a client device and a server device through a narrowband channel of a wireless data network, said method comprising:

(a) receiving a message at the client device, the message being sent from the server device to the client device through the narrowband channel of the wireless data network;

(b) preparing an acknowledgement message to be returned to the server device, the acknowledgement message including at least a portion of another message destined for the server device, said preparing (b) includes at least the operations of:

(b1) preparing a standard acknowledgement message indicating that the message has been successfully received by the client device;

(b2) determining whether there are additional messages waiting to be sent to the server device; and

(b3) modifying the standard acknowledgement message to include at least a portion of one of the additional messages, thereby producing the acknowledgement message to be returned to the server device; and

(c) sending the acknowledgement message to the server device,

wherein the standard acknowledgement message includes a user data field, and wherein said modifying (b3) operates to include at least a portion of one of the additional messages in the user data field.

2. A method as recited in claim 1, wherein the server device is an information server.

3. A method as recited in claim 1, wherein the client device is selected from the group consisting of: personal digital assistant, a mobile telephone device, or a pager, each of which has limited computing resources and a small display screen.

4. A method as recited in claim 1,

wherein the message is a SMS message, and

wherein the acknowledgement message is one of a Submit Success Report and a Delivery Success Report.

5. A method as recited in claim 1, wherein the narrowband channel is a SMS channel, and the message is an SMS message.

6. A method as recited in claim 1, wherein the additional message is a response to the message.

7. A method as recited in claim 1, wherein the additional message is unrelated to the message.

8. A method as recited in claim 1, wherein said method further comprises:

sending the message from the server device to the client device through the narrowband channel of the wireless data network.

9. A method of transmitting message packets from an initiating unit to a receiving unit over a wireless data network using a Short Message Service Center, said method comprising:

maintaining, at the receiving unit, a message queue of messages awaiting delivery;

receiving, at the receiving unit, a message from the initiating unit over the wireless communications using the Short Message Service Center;

determining whether the received message is valid;

determining whether the message queue contains a deferred message awaiting delivery to the initiating unit;

17

generating an acknowledgement message that incorporates within a user data field thereof at least a portion of the deferred message awaiting delivery to the initiating unit; and

forwarding the acknowledgement message to the initiating unit over the wireless communications using the Short Message Service Center.

10. A method as recited in claim 9, wherein the receiving unit is a server device and the initiating unit is a wireless client device.

11. A method as recited in claim 9, wherein the initiating unit is a server device and the receiving unit is a wireless client device.

12. A method as recited in claim 9, wherein the message is a SMS message, and wherein the acknowledgement message is one of a Submit Success Report and a Delivery Success Report.

13. A method as recited in claim 9, wherein the wireless data network uses a wireless communications protocol.

14. A method as recited in claim 13, wherein the wireless communications protocol is selected from a group consisting of Wireless Access Protocol (WAP) and Handheld Device Transport Protocol (HDTTP).

15. A method as recited in claim 9, wherein the acknowledgement message can incorporate up to a predetermined amount of data, and wherein when the deferred message has a size greater than the predetermined amount, the deferred message can be divided into a plurality of components each having a size not greater than the predetermined amount.

16. A method as recited in claim 9, wherein the deferred messages in the message queue are assigned priorities, and

wherein said determining of whether the message queue contains a deferred message awaiting delivery to the initiating unit operates to select the one of the deferred messages in the message queue awaiting delivery to the initiating unit based on the assigned priorities.

17. A method as recited in claim 9, wherein said wireless client device is selected from a group consisting of: personal digital assistant, a mobile telephone device, or a pager.

18. A computer readable medium including computer program code for sending messages between a client device and a server device through a channel of a wireless data network, said computer readable medium comprising:

computer program code for receiving a message at the client device, the message being sent from the server device to the client device through the channel of the wireless data network;

computer program code for preparing an acknowledgement message to be returned to the server device, the acknowledgement message including data destined for the server device, said computer program code for preparing includes at least

computer program code for preparing a standard acknowledgement message indicating that the message has been successfully received by the client device;

computer program code for determining whether there are additional messages waiting to be sent to the server device; and

computer program code for modifying the standard acknowledgement message to include at least a portion of one of the additional messages, thereby producing the acknowledgement message to be returned to the server device; and

18

computer program code for sending the acknowledgement message to the server device,

wherein the standard acknowledgement message includes a user data field, and wherein said computer program code for modifying operates to include at least a portion of one of the additional messages in the user data field.

19. A computer readable medium as recited in claim 18, wherein the data is a resource.

20. A computer readable medium as recited in claim 18, wherein the data is unrelated to the message being acknowledged by the acknowledgement message.

21. A computer readable medium as recited in claim 18, wherein the data is a reply to the message being acknowledged by the acknowledgement message.

22. A computer readable medium as recited in claim 18, wherein the client device is selected from the group consisting of: personal digital assistant, a mobile telephone device, or a pager, each of which has limited computing resources and a small display screen.

23. A computer readable medium as recited in claim 18, wherein the message is a SMS message, and wherein the acknowledgement message is one of a Submit Success Report and a Delivery Success Report.

24. A computer readable medium as recited in claim 18, wherein the additional message is a response to the message.

25. A computer readable medium as recited in claim 18, wherein the additional message is unrelated to the message.

26. A computer readable medium as recited in claim 18, wherein the channel is a narrowband channel.

27. A computer readable medium including computer program code for transmitting message packets from an initiating unit to a receiving unit over a wireless data network using a Short Message Service Center, said method comprising:

computer program code for maintaining, at the receiving unit, a message queue of messages awaiting delivery; computer program code for receiving, at the receiving unit, a message from the initiating unit over the wireless communications using the Short Message Service Center;

computer program code for determining whether the received message is valid;

computer program code for determining whether the message queue contains a deferred message awaiting delivery to the initiating unit;

computer program code for generating an acknowledgement message that incorporates within a user data field thereof at least a portion of the deferred message awaiting delivery to the initiating unit; and

computer program code for forwarding the acknowledgement message to the initiating unit over the wireless communications using the Short Message Service Center.

28. An apparatus for sending and receiving messages over a wireless data network, said apparatus comprising:

an outgoing data queue that stores data to be sent over the wireless data network;

a message manager, the message manager manages (i) the reception of incoming messages from senders over the wireless data network and (ii) the generation of outgoing messages to be sent over the wireless data network;

a storage medium that stores the incoming messages; and

a processing module operatively connected to said message manager and said storage medium, said processing module interacts with said storage medium and said

**19**

message manager to request, send and receive data over the wireless data network,  
wherein the outgoing messages generated by said message manager include acknowledgement messages that acknowledge the receipt of at least some of the incoming messages, and depending on availability of data in said outgoing data queue, the acknowledgement mes-

**20**

sages generated by said message manager include, within user data fields of the acknowledgement messages, data from said outgoing data queue destined for the respective senders of the incoming messages being acknowledged.

\* \* \* \* \*